МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

ШМАЛЬЦЕР Руслан Володимирович

Алгоритми навігації квадрокоптера з використанням інфрачервоних давачів / Quadrocopter Navigation Algorithms Using Infrared Sensors

спеціальність: 123 - Комп'ютерна інженерія
магістерська програма - Комп'ютерна інженерія

Магістерська робота

Виконав студент групи КІм-21
Р. В. Шмальцер

Науковий керівник:
д.т.н., професор, В. М. Теслюк

Магістерську роботу допущено
до захисту:

"2" 02 2018 р.

Завідувач кафедри
О. М. Березький

ТЕРНОПІЛЬ - 2018

Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії
Освітній ступінь «магістр»
спеціальність: 123 - Комп'ютерна інженерія
магістерська програма - Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ О.М. Березький
"21" 11 2016р.

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ
Шмальцер Руслан Володимирович
(прізвище, ім'я, по батькові)

1. Тема магістерської роботи «Алгоритми навігації квадрокоптера з використанням інфрачервоних давачів/ Algorithms for navigating the quadrocopter using infrared sensors»
керівник роботи д.т.н., професор Теслюк В.М.
затверджені наказом по університету від 23 жовтня 2017 р. № 1325.

2. Строк подання студентом роботи «15» січня 2018 року

3. Вихідні дані до магістерської роботи

Об'єкт дослідження – процес передачі даних від оптичних сенсорів та модуля позиціонування та їх подальша обробка.

Предмет дослідження – спеціалізовані компоненти для сиситем навігації.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
- дослідити відомі системи глобального позиціонування;
- дослідити алгоритми навігації;
- розробити метод і алгоритми , що приведуть до створення програмного модуля;
- створити програмно-апаратне забезпечення для навігації квадрокоптера з використанням інфрачервоних давачів;
- виконати тестування розробленого програмно-апаратного забезпечення.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
- тема, мета, завдання, методи досліджень, наукова новизна, практичне значення;
- актуальність;

– класифікація програмного забезпечення;
– об'єкт дослідження.

6. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Антиплагіат	Мельник Г.М., доцент		
Нормо-контроль	Гураль І. В., викладач		

7. Дата видачі завдання «21» 11 2016 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Приміт
1	Аналіз методів та алгоритмів роботи систем глобального позиціонування	3.11.2016 – 1.01.2017	
2	Аналіз алгоритмів навігації	2.01.2017 – 31.05.2017	
3	Система навігації квадрокоптера з використанням інфрачервоних давачів	1.06.2017 – 25.01.2018	
4	Нормоконтроль, попередній захист	16.01.2018 – 2.02.2018	
5	Захист	4.02.2018	

Студент _____ (підпис) Шмальцер Р. В.

Керівник магістерської роботи _____ (підпис) д.т.н., професор Теслюк В.М.

ЗМІСТ

ВСТУП

Алгоритми навігації роботизованої системи з використанням інфрачервоних давачів відіграють ключову роль системах навігації безпілотних апаратів тому вони виступають важливими об’єктоми дослідження у сфері інформаційних технологій.

Для дослідження системи навігації, проблем розподілення навантаження - інформаційні технології використовують весь свій методологічний інструментарій. Сенсорні системи є основним джерелом інформації для автоматизованих систем, дозволяючи останнім обробляти отримані дані про навколишній світ, контролювати виконання функцій, оптимізувати власну структуру. Розробка оптимальніших організацій сенсорних систем є важливим напрямом у вдосконаленні автоматизованих систем. Робототехніка, яка займається побудовою таких систем, у наш час залишається перспективною та актуальною.

У рамках проекту планується побудова системи управління інфрачервоними сенсорами та системою GPS як окремого блоку квадрокоптера. При цьому цей блок буде досить гнучким за своїми фізичними властивостями.

Для створення прототипу мобільної рухомої платформи для робота вибрано такі компоненти: Плата MultiWii, що є основою розробки, оснащена мікро-контролером Atmega328, дає змогу під’єднувати до себе і здійснювати управління різними фізичними пристроями, такими як, датчики відстані, GPS тощо.

Метою дослідження є розробра алгоритму навігації роботизовниї системи на основі інфрачервоних сенсорів відстані та використання методу SLAM.

Обєктом дослідження є процес передачі даних оптичних сенсорів та модуля позиціонування та їх подальша обробка від. Предметом дослідження є спеціалізовані компоненти для сиситем навігації.

Досягнення поставленої мети реалізовано з використанням методів

інтелектуального аналізу даних отриманих за допомогую давачів.

Наукова новизна дипломної роботи полягає в тому, що було розроблено та відлагоджено алгоритм, що дозволяє фільтрувати шуми від інфрачервоних сенсорів відстані за допомогою фільтру Калмана.

Актуальність теми дослідження полягає в тому, що визначення положення мультироторних систем та перешкод вимагає точних та якісних результатів роботи контролера польоту.

1 АНАЛІЗ МЕТОДІВ, АЛГОРИТМІВ ТА ПРОГРАМНО-АПАРАТНИХ ЗАСОБІВ НАВІГАЦІЙНИХ СИСТЕМ КВАДРОКОПТЕРІВ

1.1 Аналіз існуючих рішень, моделей, алгоритмів, та фізичних реалізацій

Система навігації які використовуються для роботизованих платформ – це комплексні електронно-технічні системи, що складається з сукупності наземного і космічного устаткування, призначена визначення місцеположення (географічних координат і висоти), і навіть параметрів руху (швидкості та направлення руху, і т.д.) для наземних, водних і повітряних об'єктів.

Алгоритм роботи супутникових систем навігації заснований на вимірюванні відстані від антени на об'єкті, географічні координати якого необхідно отримати, до супутників, місцезнаходження яких у будь-який момент відоме з великою точністю. Таблиця положень усіх супутників (так званий альманах) має бути в кожному приймачі супутникового сигналу до початку вимірювань. Кожний супутник передає в своєму сигналі весь альманах. Таким способом, знаючи відстані до декількох супутників системи, за допомогою звичайних геометричних побудов на основі альманаху можна вирахувати положення об'єкта в просторі.

Метод вимірювання відстані від супутника до антени приймача базується на визначеності швидкості поширення радіохвиль. Для реалізації можливості вимірювання часу радіосигналу, що поширюється, кожний супутник навігаційної системи випромінює сигнали точного часу, використовуючи системну синхронізацію. При роботі супутникового приймача його годинник синхронізується з системним часом і при подальшому прийманні сигналів обчислюється затримка між часом випромінювання, що міститься в самому сигналі, і часом приймання сигналу. Маючи цю інформацію, навігаційний приймач вираховує координати антени. Вся решта параметрів руху (швидкість, курс, пройдена відстань) обчислюється на основі вимірювання часу, який об'єкт витратив на переміщення між двома або більше точками з координатами, визначеними попередніми обчисленнями.

На даний час основними навігаційними системами є :

- NAVSTAR
- ГЛОНАС
- Galileo
- IRNSS
- QZSS

Основою системи NAVSTAR (Navigation Satellite Time and Ranging) є 24 супутники, що працюють у єдиній мережі й обертаються на шести різних кругових орбітах, розташованих під кутом 60° одна до одної. На кожній орбіті розміщено по 4 супутники, висота орбіт приблизно дорівнює 20 200 км а період обертання кожного супутника навколо землі дорівнює 12 годинам. Таким чином, із будь-якої точки земної поверхні зазвичай одночасно видно від чотирьох до дванадцяти таких супутників (рисунок 1.1).

Супутники перебувають під контролем станцій, які розташовані на Землі. Розміщуються такі станції на Колорадо-Спрінгз, Дієго-Гарсія, острові Вознесіння, атолі Квадженлейн і на Гаваях. Вся інформація, що проходить через ці станції, записується ними та передається на головну станцію на військовій базі Falcon (штат Колорадо).

GPS-приймач обчислює власне місцезнаходження, вимірюючи час проходження сигналу від GPS-супутників. Кожен супутник постійно надсилає повідомлення, в якому міститься інформація про час, точку орбіти супутника, з якої було надіслано повідомлення (ефемерида), та загальний стан системи й приблизні дані орбіт усіх інших супутників системи GPS (альманах).

Ці сигнали розповсюджуються зі швидкістю світла в космосі (і з трохи меншою швидкістю — в атмосфері). Приймач визначає час затримки в надходженні сигналу та обчислює відстань до супутників, виходячи з якої, застосувавши метод трилатерації, визначає своє місце. Отримані координати перетворюються в наочну форму (широта та довгота чи положення на карті) та відображаються користувачеві.

Теоретично для визначення власних координати достатньо визначити відстань до трьох супутників. Однак для обчислення положення необхідно знати час із високою точністю. Щоб усунути потребу в високоточному годиннику, отримують інформацію з 4-х чи більше супутників, тобто, GPS-приймач використовує чотири параметри для обчислення чотирьох невідомих: x, y, z та t.

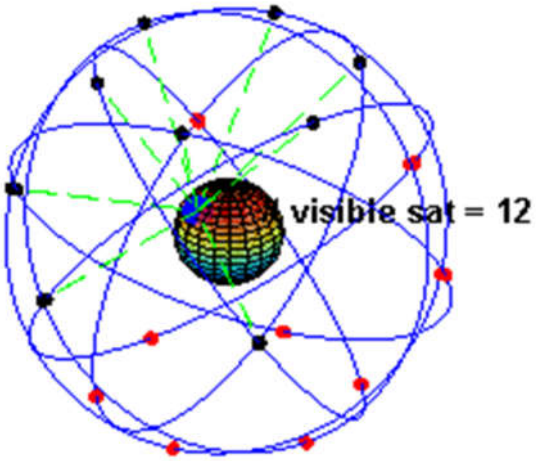

Рисунок 1.1 – Орбіти супутників системи GPS.

У деяких окремих випадках можна обійтися меншою кількістю супутників. Якщо заздалегідь відома одна змінна (наприклад, висота над рівнем моря човна в океанідорівнює 0), приймач може обчислити положення, використовуючи дані з трьох супутників. Також на практиці приймачі використовують різну допоміжну інформацію для обчислення положення з меншою точністю в умовах відсутності одразу чотирьох супутників.

Глонасс (Глобальна Навігаційна Супутникова Система) — радянська російська радіонавігаційна супутникова система, розроблена на замовлення Міністерства оборони СРСР. Одна із двох функціонуючих на сьогодні глобальних систем супутникової навігації. Розгортання системи у космосі зроблено за допомогою супутників «Глонасс-К» та «Глонасс-М» (Глонасс 2-го покоління).

Основою системи є 24 супутника, що обертаються над поверхнею Землі в трьох орбітальних площинах. Координати визначаються за принципом, узятим за аналогією американської системи глобального позиціонування GPS. Як альтернатива обом системам у Європі розробляється система Галілео.

Супутники системи ГЛОНАСС стало розповсюджують (передають) радіовипромінювання двох типів: навігаційний сигнал СТ діапазону L1 (1,6 ГГц) та навігаційний сигнал високої точності ВТ діапазонів L1 и L2 (1,2 ГГц).

Супутники ГЛОНАСС перебувають на радіальній орбіті на середній висоті 19400 км із нахилом 64,8° і періодом в 11 годин 15 хвилин. Така орбіта більше придатна для застосування на високих широтах (північний і південний полярний регіон), де сигнал NAVSTAR приймається погано. Група супутників розгорнута в трьох орбітальних площинах, із 8 рівномірно розподіленими супутниками в кожній. Для створення глобального покриття необхідно 24 супутника, а для покриття території Росії необхідно 18 супутників. Сигнали передаються з направленістю в 38° з використанням правої кругової поляризації, із потужністю 316—500 Вт (EIRP 25-27 dBW).

Для визначення координат приймач повинен отримувати сигнал як мінімум від чотирьох супутників і розрахувати відстань до них. При використанні трьох супутників визначення координат ускладнене через помилки, що викликані неточністю годинника приймача.

Галіле́о (англ. Galileo) — проект супутникової системи навігації Європейського Союзу та Європейського космічного агентства,— як альтернатива американській системі GPS та російській ГЛОНАСС. Проект вартістю 2 мільярда доларів названий в честь Італійського астронома Галілео Галілея.

Європейська система призначена для вирішення навігаційних завдань для будь-яких рухомих об'єктів з точністю менше одного метра. Крім країн європейського співтовариства, досягнуті домовленості на участь у проекті з державами — Китай, Ізраїль, Південна Корея й Україна. Крім того, ведуться переговори з представниками Аргентини, Австралії, Бразилії, Чилі, Індії,

Малайзії та Росії. Очікувалося, що Галілео стане до ладу в 2013, коли на орбіту буде виведено всі 30 запланованих супутників (27 операційних і 3 резервних) і космічний сегмент буде доповнений наземною інфраструктурою, що має два центри управління й глобальну мережу передавальних і приймальних станцій.

На відміну від американської системи GPS і російської ГЛОНАСС, система Галілео не контролюється національними військовими відомствами, однак, у 2008 році парламент ЄС ухвалив резолюцію «Значення космосу для безпеки Європи», згідно з якою допускається використання супутникових сигналів для військових операцій, що проводяться в рамках європейської політики безпеки. Розробку системи здійснює Європейське космічне агентство. Загальні витрати оцінюються в 4,9 млрд євро.

Quasi-Zenith Satellite System (QZSS, «Квазізенітна супутникова система») - проект трьохсупутникової регіональної системи синхронізації часу і одна з QZSS може покращити работу системи GPS двома способами: підвищенням доступності GPS-сигналів, підвищенням точності та надійності работи навігаційних систем, працюючих з GPS. Систем диференціальної корекції для GPS, сигнали якої будуть доступні в Японії. Перший супутник Мітібікі був запущений 11 вересня 2010. Повне розгортання системи передбачалось в 2013р.

Робота над спільним проектом квазі-зенітної супутникової системи була схвалена урядом Японії в 2002 році. У неї включилися компанії Advanced Space Business Corporation (ASBC), Mitsubishi Electric Corp., Hitachi Ltd. і GNSS Technologies Inc. Однак ASBC припинила існування в 2007 році.

Робота була продовжена організацією Satellite Positioning Research and Application Center (SPAC). SPAC належить чотирьом департаментам уряду Японії: міністерствам освіти, культури, спорту, науки і технологій; внутрішніх справ і зв'язку. Міністерство економіки, торгівлі і промисловості і міністерству землі, інфраструктури, транспорту та туризму.

QZSS призначена для мобільних додатків, для надання послуг зв'язку (відео, аудіо та інші дані) і глобального позиціонування. Що стосується послуг позиціонування, QZSS сама по собі надає обмежену точність і за існуючої

специфікації не працює в автономному режимі. З точки зору користувачів QZSS постає як система диференціальної корекції. Система позиціонування QZSS може працювати спільно з геостаціонарними супутниками в японській системі MTSAT, що знаходиться в процесі створення, яка сама по собі є системою диференціальної корекції, подібній системі WAAS, створеної США.

Супутники перебуватимуть на високій еліптичній орбіті «Тундра». Такі орбіти дозволяють супутнику триматися більше 12 годин на день з кутом піднесення більше 70 ° (тобто більшу частину часу супутник знаходиться практично в зеніті). Цим і пояснюється термін «quasi-zenith», тобто «удаваний знаходяться в зеніті», який дав назву системі. Станом на червень 2003 року пропоновані орбіти розташовуються в діапазоні від 45 ° з невеликим ексцентриситетом та до 53 ° зі значним ексцентриситетом.

1.2 Аналіз методів і алгоритмів аналізу отриманих даних з давачів

Давач (сенсор) - конструктивно відособлений первинний вимірювальний перетворювач, від якого поступають сигнали вимірювальної інформації. Але первинний перетворювач може знаходитися у вимірювальному ланцюзі будь-якого засобу вимірів і не обов'язково має бути сенсором, тобто конструктивно відособленим.

Таким чином, під давачем, сенсором слід розуміти конструктивно відособлену сукупність первинних вимірювальних перетворювачів, що сприймає одну або декілька вхідних величин і що перетворює їх у вимірювальні сигнали. Слід сказати, що для виконання функцій сприйняття вхідних величин використовують також вимірювальні установки, що складаються із складної сукупності елементів і складні в реалізації фізичні ефекти.

Поняття сенсор необхідно відрізняти від поняття перетворювач. Перетворювач конвертує один тип енергії в інший, тоді як сенсор перетворить певний тип енергії зовнішньої дії в електричний сигнал.

У вимірювально-інформаційній техніці за кордоном використовують декілька термінів, що відбивають особливості виконання функцій сприйняття і формування вимірювальних сигналів: sensor, gange (англ.), Geber, Primarmessumformer, Transducteur de mesure (франц.). Нині склалося таке положення, що термін "сенсор" використовується для позначення вимірювального перетворювача, який виконує функції сприйняття вхідної величини і формування вимірювального сигналу.

Принципи роботи сенсорів. Сучасні сенсорні системи, як правило, працюють одночасно з набором різноманітних сенсорів, таких як системи технічного зору, оптичні, ультразвукові, тактильні, силові і інші типи сенсорів. Основна вимога, яка пред'являється до цих систем, являється те, що обробка зорової, слухової, тактильної і іншій інформації повинна здійснюватися в реальному масштабі часу.

Сенсорні системи дозволяють виконувати технологічні і інші операції, використовуючи зворотний зв'язок, аналогічний тій, яка має місце при роботі людини.

На самому верху знаходиться людино-машинний інтерфейс, що посилає команди супервізору, який, у свою чергу, посилає команди усім пристроям та модулям. Це модулі програмного забезпечення, кожен з яких узгоджений із спеціальною функцією машини (робота).

Визначення положення фізичних об'єктів і їх переміщень є важливою функцією багатьох автоматизованих систем. Вона потрібна практично для усіх АСУТП, систем управління навігацією безпілотними апаратами, охоронних систем та іншим системам.

Для виявлення небезпечних відстаней між об'єктами зазвичай застосовуються детектори зближення. Такі детектори, по суті, є пороговими пристроями, реалізованими на базі сенсорів положення об'єкту. Сенсори положення - це, як правило, лінійні пристрої, вихідні сигнали яких відповідають відстані між об'єктом і опорною точкою. Детектори зближення є простішими пристроями, сигнали на виході яких з'являються тільки у разі

виявлення критичної відстані до об'єкту.

Сенсори швидкості і прискорення. До складу усіх акселерометрів входить спеціальний елемент, званий інерційною масою, рух якого відстає від руху корпусу. І незалежно від конструкції сенсора прискорень його основна мета полягає в детектуванні переміщення цієї маси відносно корпусу пристрою і перетворенні його в пропорційний електричний сигнал. Тому іншою складовою частиною усіх акселерометрів є детектор переміщень, здатний вимірювати мікроскопічні амплітуди вібраційних коливань або лінійних прискорень. Місткістю метод перетворення переміщень в електричний сигнал є найперевіренішим і надійнішим, але в таких сенсорах завжди необхідно компенсувати дрейф різних параметрів, а також пригнічувати всілякі перешкоди. Тому зазвичай акселерометри мають диференціальну структуру, для чого до їх складу вводиться додатковий конденсатор, ємність якого має бути близька до ємності основного конденсатора.

Оптичний інфрачервоний далекомір Sharp 2y0a21 (рисунок 1.2) використовується для виміру відстані від сенсора до об'єкту. Для використання далекоміра потрібно зібрати макет (підключити до контроллера, живлення і т.п). Потім потрібно створити програму, що керує ним. У кінці необхідно направити сенсор на об'єкт під кутом не більше 15° і на регламентованій відстані, та починати роботу. Управління та збір даних з даного сенсора здійснюється за допомогою польотного контролера або від іншого мікропроцесорного пристрою, що керує через відповідні інтерфейси, та входить в систему контролера польоту.

Рисунок 1.2 – Оптичний інфрачервоний далекомір Sharp 2y0a21

Характеристики:

- робоча напруга: від 4.5 до 5.5 В;
- робочий струм: 30 mA;
- робоча частота: 40 кГц;
- максимальна дальність виміру : 80 см;
- мінімальна дальність виміру : 10 см;
- габарити: 37 х 15 мм;
- вага: 7 г;

Принцип роботи: Імпульси ІЧ випромінювання випускаються випромінювачем. Це випромінювання поширюється і відбивається від обєктів що знаходяться в полі зору сенсора. Відбите випромінювання повертається на приймач. Згенероване випромінювання і відбиті промені утворюють трикутник «випромінювач - обєкт відбиття - приймач».

Кут відбиття безпосередньо залежить від відстані до об'єкта. Отримані відбиті імпульси збираються високоякісною лінзою і передаються на лінійну CCD матрицю. За засвітленю певною частиною ділянки CCD матриці визначається кут відбиття і вираховується відстань до об'єкта (рисунок 1.3).

Цей алгоритм роботи більш захищений від ефектів інтерференції випромінювання і різної відбивної здатності оптичних поверхонь, виконаних з різних матеріалів і пофарбованих у різні кольори. Наприклад, стало можливе визначення чорної стіни при яскравому освітленні.

Позначення контактів: Vcc (висновок для живлення +5В), Output (вихід), GND (загальний вивід). Живлення модуля здійснюється або від контроллера (іншого мікропроцесорного пристрою, що керує), або від зовнішнього джерела живлення (блоку живлення, батареї).

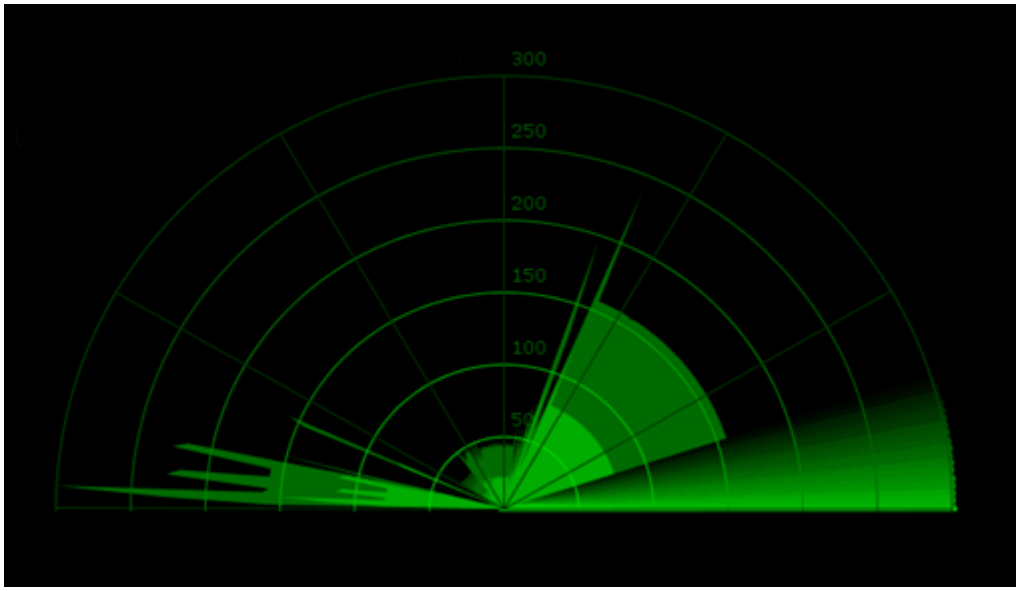

Рисунок 1.3 – Результат обробки даних від іфрачервоних давачів.

GPS-модуль використовується для отримання географічних координат розташування антени приймача за допомогою GPS (системи глобального позиціонування) і їх передачі на контролер. Модуль може бути використаний в пристроях на Arduino, AVR, PIC, ARM. Практичне застосування: позиціонування роботів, безпілотних літальних апаратів, метеорологічних датчиків і т.п.

Для використання GPS-модуля спочатку потрібно підключити його до живлення і контролера. Після того, як модуль виявить достатню кількість супутників, повинен загорітися сигнальний світлодіод. Після цього в контролер

потрібно записати програму для роботи з GPS. Модуль видає результати у форматі NMEA. Для роботи з даними, які передає модуль і для настройки можна використовувати програму від виробника U-Center (рисунок 1.4 – 1.5).

Управління GPS модулем можемо здійснюватися або з контролера по інтерфейсів (UART, SPI, DDC, IIC) або з комп'ютера по інтерфейсу USB.

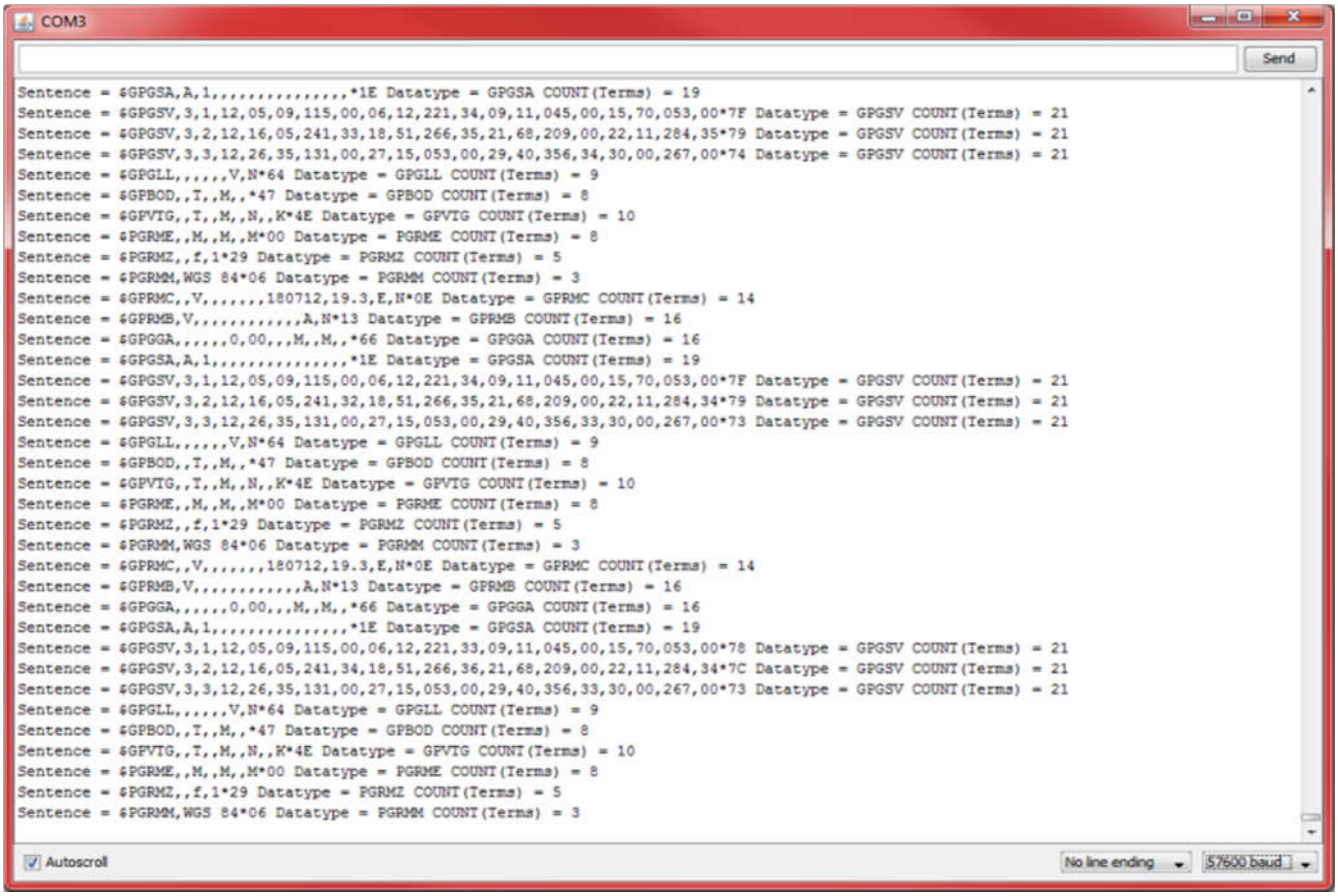

Рисунок 1.4 – Дані які надходять з GPS модуля у форматі NMEA.

Протокол NMEA 2000 описує не тільки дані, отримані з GPS-приймачів, але і вимірювання сонарів, радарів, електронних компасів, барометрів і інших навігаційних пристроїв, що використовуються для навігації. Інтерфейс обміну даними більшості портативних GPS-приймачів реалізований відповідно до NMEA-специфікацією. Більшість навігаційних програм, які забезпечують відображення даних в реальному часі, підтримують NMEA протокол. Ці дані містять повні навігаційні вимірювання GPS-приймача — позицію, швидкість і час.

Повна специфікація NMEA повідомлень відсутня у вільному доступі і її не можна офіційно скачати в електронному вигляді. Окремі її розділи, загальний опис NMEA протоколу і найбільш популярних повідомлень можна знайти в Інтернеті.

Усі NMEA повідомлення складаються з послідовного набору даних, розділених комами (рисунок 1.4). Кожне окреме повідомлення не залежить від інших і є повністю «завершеним». NMEA Повідомлень включає: заголовок, набір даних, представлених ASCII символами, поле «контрольної суми» для перевірки достовірності переданої інформації.

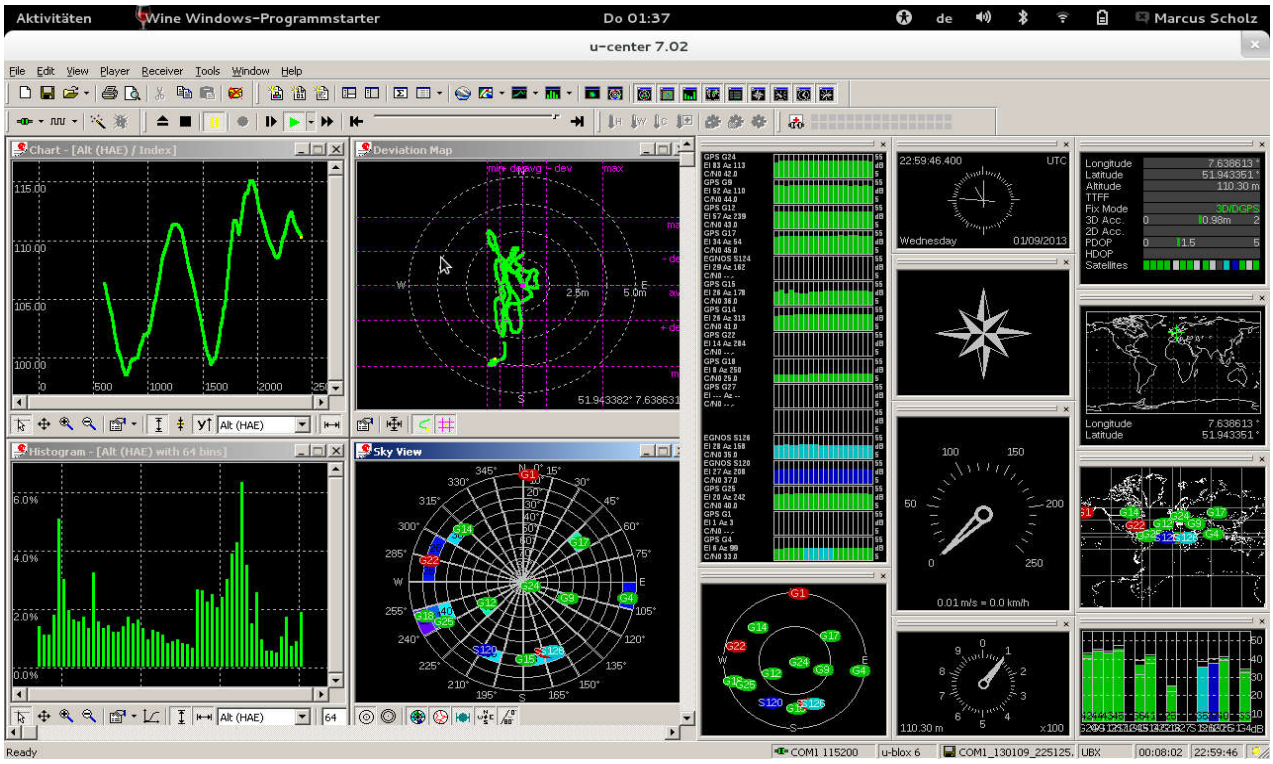

Рисунок 1.5 – Обробка даних з GPS модуля з використанням U-Center.

Як правило, заголовок складається з п'яти символів. Перші два символи визначають тип повідомлення, а решту три - його назва. Наприклад, заголовок GPS NMEA повідомлень починається з «GP». Повідомлення, які не описані в специфікації NMEA, але реалізовані в GPS-приймачах відповідно до загальних правил, мають префікс «P», доповнений трьома символами, унікальними для

кожної компанії. Наприклад, NMEA повідомлення Garmin мають префікс «PGRM», Magellan - «PMGN».

Кожне NMEA повідомлення починається з «$», закінчується «\n» (перенос рядка) і не може бути довшим 80 символів. Всі дані містяться в одному рядку і відокремлені один від одного комами. Інформація представлена у вигляді ASCII тексту і не вимагає спеціального декодування. Якщо дані не вміщаються в виділені 80 символів, то вони «розбиваються» на кілька повідомлень. Такий формат дозволяє не обмежувати точність і кількість символів в окремих полях даних. Наприклад, дрібна частина значення координат може бути представлена трьома або чотирма знаками після коми, але це ніяк не повинно вплинути на роботу програмного забезпечення, які виділяє потрібні дані з повідомлення за номером поля.

Поле “контрольної суми”. В кінці кожного NMEA повідомлення міститься поле “контрольної суми”, відокремлене від даних символом «*». При необхідності воно може використовуватися для перевірки цілісності та достовірності кожного прийнятого повідомлення.

Вхідні повідомлення NMEA. Протокол NMEA 2000 підтримує не тільки вихідні, а й вхідні повідомлення, за допомогою яких, наприклад, можна оновити або додати шляхові точки маршруту. Ці повідомлення повинні бути сформовані в суворій відповідності з форматом NMEA, в іншому випадку, вони будуть проігноровані GPS-приймачем.

1.3 Програмно-апаратні засоби навігаційних систем безпілотних літальних апаратів

Для навігації мобільного робота було вибрано такі компоненти (таблиця 1.1).

Таблиця 1.1 – Перелік компонентів

Компонент	Кількість
MultiWii V2.5	1
Оптичний сенсор Sharp 2y0a21	3
Ublox GPS	1

Плата MultiWii V2.5 (Рисунок 1.6), що є основою розробки, оснащена мікро-контролером Atmega328P, дає змогу під’єднувати до себе і здійснювати управління різними фізичними пристроями, такими як, наприклад, сенсори відстані, випромінювачі тощо.

MultiWii - це ефективний засіб розробки програмованих електронних пристроїв, які, на відміну від персональних комп'ютерів, орієнтовані на тісну взаємодію з навколишнім світом. MultiWii - це відкрита програмована апаратна платформа для роботи з різними фізичними об'єктами і є простою платою з мікроконтроллером, а також спеціальне середовище розробки для написання програмного забезпечення мікроконтроллера.

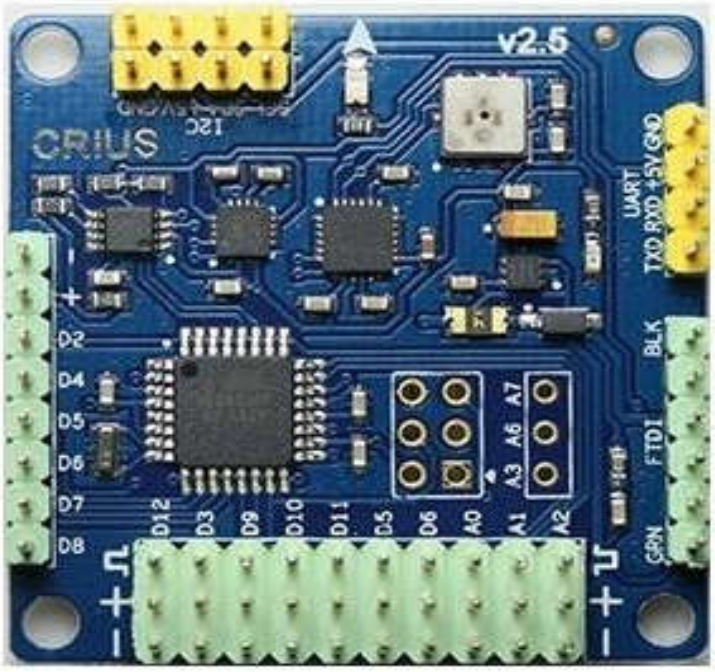

Рисунок 1.6 - Плата MultiWii V2.5.

MultiWii може використовуватися для розробки інтерактивних систем, керованих різними сенсорами та системами контролю. Такі системи, у свою чергу, можуть керувати роботою різних індикаторів, двигунів і інших пристроїв. Проекти MultiWii можуть бути як самостійними, так і взаємодіяти з програмним забезпеченням, що працює на персональному комп'ютері.

MultiWii також спрощує процес роботи, але на відміну від інших систем надає ряд переваг:

- Низька вартість. В порівнянні з схожими апаратними платформами, плати MultiWii мають відносно низьку вартість: готові модулі MultiWii коштують 80$.
- Кроссплатформенність.
- Просте і зручне середовище програмування.
- Розширюване програмне забезпечення з відкритим початковим кодом. Програмне забезпечення має відкритий початковий код, завдяки цьому досвідчені програмісти можуть змінювати і доповнювати його. Можливості можна також розширювати за допомогою C бібліотек. Алгоритм написаний на мові AVR C, кваліфіковані користувачі, що бажають розібратися в технічних деталях, можуть легко модифікувати програмне забезпечення.
- Розширюване відкрите апаратне забезпечення.

Платформа для MultiWii (Рисунок 1.7) робот – мобільна чотирьохроторна плат-форма з 4 гвинтами Multi-Quad-X.

Рисунок 1.7 – Платформа Multi-Quad-X.

За допомогою оптичного сенсора відстані Sharp 2y0a21 Module (Рисунок 1.2), робот зможе реагувати на приближення до предметів навколишнього середовища.

Ця система знаходить застосування при оминанні перешкод квадрокоптерами та іншими мальтироторними системами.

1.4 Постановка задачі

Метою дипломного проекту є розробка алгоритму системи навігації для квадрокоптера на основі ІЧ давачів та програмно-апаратних модулів управління Оптичними ІЧ сенсорами на базі контролера польотів MultiWii V2.5 та GPS модуля UBlox.

Для досягнення мети вирішуються наступні задачі:

– Огляд відомих алгоритмів та платформ мобільних роботизованих платформ.
– Огляд існуючих апартних засобів та сенсорних пристроїв мобільних роботів.
– Розробка алгоритму системи управління роботом.
– Розробка алгоритму управління оптичними сенсорами.
– Розробка алгоритму управління системою навігаці.

- Розробка схеми взаємозв’язку програмних модулів.
- Програмна реалізація основних модулів управління оптичними ІЧ та GPS сенсорами та взаємодію з контролером польоту.
- Проведення експериментальних досліджень та симуляції.

Розгляд задач містяться в наступних розділах проекту.

2 АЛГОРИТМИ ФУНКЦІОНУВАННЯ СИСТЕМИ НАВІГАЦІЇ

2.1 Метод SLAM

При вирішенні задач навігації роботів, одночасна локалізація і картографування (SLAM) є алгоритмічною обчислювальною задачею побудови і оновлення мапи невідомого оточення з одночасним відстежуванням місцезнаходження рухаючись по ньому. Популярні методи вирішення включають в себе фільтр часток і розширений фільтр Калмана.

Алгоритми SLAM обмежуються наявними ресурсами, таким чином не можуть бути абсолютно досконалими, бо досягають оперативної доступності.

$$P(m_t, x_t | o_{1:t})$$

До статистичних методів, що використовуються для задач апроксимації наведених вище, відносяться: фільтр Калмана, фільтр часток (він же Метод Монте-Карло) і узгоджене сканування діапазонних даних. Вони дозволяють визначити оцінку функції апостеріорної ймовірності для позиції робота і параметрів мапи. Техніки оцінювання приналежності до множини в основному засновуються на поширенні сталого інтервалу. Вони надають множину, яка містить позицію робота і множину апроксимації мапи. "Bundle adjustment" є наступною популярною технікою? що використовується для SLAM, який використовує дані зображень, яка одночасно визначає позиції робота і орієнтирів місцевості, підвищуючи точність мапи, і навіть використовується в коменрційних SLAM системах, таких як Проект Tango компанії Google.

Нові алгоритми SLAM досі потребують активного дослідження і пошуку, і часто обумовлений різними вимогами і припущення щодо типів карт, датчиків і моделей. Більшість SLAM систем можна розглядати як комбінації виборів кожного з цих аспектів.

Побудова топологічних мап це метод представлення довколишнього світу, який узагальнює структуру (тобто, топологію) оточення замість створення геометрично точної мапи. Топологічні методи SLAM використовуються для підвищення загальної узгодженості метричних алгоритмів SLAM.

На противагу тому, для топологічного представлення світу існують мапи у вигляді поверхонь, які використовують масиви оцифрованих елементів (зазвичай квадратних, або гексагональних клітин), і роблять припущення яка клітина зайнята в конкретний момент. Зазвичай клітини вважаються статистично незалежними, аби спростити обчислення.

$$P(m_t|x_t, o_{1:t}) = \sum_{x_t} \sum_{m_t} P(m_t|x_t, m_{t-1}, o_t) P(m_{t-1}, x_t|o_{1:t-1}, m_{t-1})$$

Існуючі інерціальні системи навігації та системи, засновані на застосуванні GPS з об'єктивних причин не в состоя ванні забезпечити необхідну точність визначення поточного місцезнаходження автономного мобільного робота, в той час як розвиток технологій локальної навігації на основі візуальної зворотнього зв'язку в комплексі з застосуванням високо кокачественних систем технічного зору дозволяє з високим ступенем досто вірності здійснювати оцінку не тільки координат самого робота, а й навко лишнього його об'єктів.

Картографування місцевості автономними мобільними роботами перед-
ставлять собою комплексну проблему, можливість вирішення якої знаходиться
в безпосередній залежності від якості наявних інформаційно вимірювальних
засобів, а також умов навколишнього середовища. Коло завдань, потребуючих одночасного вирішення проблеми локалізації та побудови карти в умовах недосконалих інформаційно-вимірювальних засобів, носить загальний назва SLAM (Simultaneous Localization And Mapping). В даний час існує кілька основних підходів до вирішення цих завдань: розширений фільтр Калмана (extended Kalman filter, EKF), FastSLAM, DP-SLAM.

Основним недостатком у даного підходу є квадратична залежність складності алгоритму від кількості спостережуваних орієнтирів (практично не перевищує нескольких сотень орієнтирів). В даний час існує і активно розвивається альтернатівний підхід, названий FastSLAM, в основі якого лежить так званим мий фільтр частинок (Particle Filter, Monte Carlo methods). На відміну від EKF в FastSLAM одна велика карта розглядається як сукупність

локальних підкарт, що дозволяє прибрати залежність орієнтирів один від одного і таким чином значно скоротити час перерахунку оцінки стану системи.

Проте у кожного з цих методів є свої обмеження і недоліки, що ще раз наголошує на необхідності вдосконалення алгоритмів картографії місцевості автономними мобільними роботами.

SLAM - це не якийсь певний алгоритм або набір ПО - це концепція, загальна методологія для вирішення двох завдань:

1) побудова карти дослідженого простору;

2) побудова траєкторії руху робота на мапі.

Треба відзначити що повністю дана задача як і раніше не вирішена і до сих пір ведуться дослідження.

Перший принциповий момент - SLAM не припускає будь-яких знань про середовище - ні міток на місцевості ні попередньої карти немає - всі рішення будуються тільки на результатах вимірювань датчиків (зазвичай це rgb- і далекомірна камери, але буває і додаткове обладнання гіроскопа, gps-датчика і т.д.).

Другий не менш важливий момент - середовище вважається статичним. Це означає, що якщо робот проїхав повз фікуса, ніхто за його спиною цей же самий фікус рухати не стане. В кадрі - в розглядаємому сенсорами просторі - жодного стороннього руху немає, ніхто в кадрі не ходить. Освітлення радикально не змінюється - тобто тіні по стінах не стрибають. Такі умови складно гарантувати в природному середовищі - приємно шелестить листя або цікава кішка запросто може дезорієнтувати робота в просторі. Проте можна відсікати динамічну частину та робити поправку на неї. У загальному випадку SLAM можна описати як повторює послідовність кроків:

1) сканування навколишнього простору;

2) визначення зміщення на основі порівняння поточного кадру з попереднім;

3) виділення на поточному кадрі особливостей, міток;

4) зіставлення міток поточного кадру з мітками отриманими за всю

історію спостережень;

5) оновлення на основі цієї інформації положення робота за всю історію спостережень;

6) перевірка на петлі - не проходим ми повторно по одній і тій же місцевості;

7) вирівнювання загальної карти світу (відштовхуючись від положення міток і робота за всю історію спостережень).

На кожному кроці алгоритму ми маємо в своєму розпорядженні припущеннями про структуру світу (у нас є карта) і історією руху робота в ньому (траєкторія руху щодо карти з розбивкою за часом).

Представлені реалізації поділяються на дві великі групи - SLAM frontend і SLAM backend.

Завдання SLAM-fronted'a зводиться до оцінки просторових відносин між особливостями сцени і позами робота з деяким значенням ймовірності. SLAM-fronted - набір методів для перетворення отриманих даних від сенсорів до уніфікованого поданням (структурі спеціального виду - див. Далі) подається на вхід безпосередньо SLAM-ядру. В даний час, для цих цілей використовуються графи спеціального виду (варіюються від зваженого, заснованого на результатах вимірювань, до динамічної мережі Байєса, щосили відштовхується від ймовірнісної моделі положення робота в світі). Даний компонент специфічний для робота (які там у нього сенсори, як організовано перетворення вхідної інформації і т.п.), а тому якоїсь універсальної бібліотеки, що реалізує весь функціонал я не зустрічав. Хоча список підтримуваних сенсорів того ж ROS - robot operaring systems вражає. Принципи вибору особливостей на кожному кадрі тут описані не будуть - хто цікавиться гуглити по SIFT, SURF, NARF, FAST і т.д.

$$P(x_t|o_{1:t}, m_t) = \sum_{m_{t-1}} P(o_t|x_t, m_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}|m_t, o_{1:t-1})/Z$$

SLAM-ядро - безпосередня реалізація рішення задачі SLAM - оптимізація отриманих від фронтенда просторових відносин з метою максимізації

ймовірності. Дане завдання абстрагування від того, яким чином отримані вхідні дані - головне, щоб формат уявлення (граф, мережа, матриця, список і т.д.) інформації про середовище і положенні робота в ньому співпадали. Наприклад, в якихось випадках необхідна висока точність і не принципово час обчислень, в інших випадках критичний швидкий результат. Якщо ми зберігаємо всі наші переміщення і виміри у вигляді спеціального графа то ми в першому випадку згодуємо цей граф на оптимізацію iSam'у, а в другому віддамо TORO'у.

2.2 Розширений фільтр Калмана

У теорії статистичного оцінювання розширений фільтр Калмана — це нелінійна версія фільтру Калмана, що лінеаризується навколо оцінки поточного середнього значення та коваріації. У випадку гарно визначених моделей переходу розширений фільтр Калмана було визнано стандатом де-факто у теорії оцінювання нелінійних станів, навігаційних системах та GPS.

Праці, що встановлюють математичні основи фільтрів типу фільтра Калмана, було опубліковано між 1959 та 1961 роками. Фільтр Калмана є оптимальною оцінкою для моделей лінійних систем з додатковим незалежним білим шумом як у системі переходу, так і в системі вимірювання. На жаль, більшість систем у техніці є нелінійними, тому одразу було зроблено певну спробу застосування цього методу фільтрування до нелінійних систем. Більшість цієї роботи було зроблено у дослідницькому центрі Еймса НАСА. Розширений фільтр Калмана пристосував прийоми з числення, а саме багатовимірні розклади в ряди Тейлора, для лінеаризації моделі навколо робочої точки. Якщо модель системи (як описано нижче) не достатньо добре відома, або є неточною, то для оцінювання застосовуються методи Монте-Карло, зокрема частинкові фільтри. Методи Монте-Карло передують існуванню розширеного фільтру Калмана, але є обчислювально витратнішими для будь-якого простору станів з помірною кількістю вимірів.

У розширеному фільтрі Калмана моделі переходу стану та спостереження не повинні бути обов'язково лінійними функціями стану, натомість вони можуть бути нелінійними диференційовними функціями.

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{w}_{k-1}$$
$$\boldsymbol{z}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k$$

де wk та vk є шумами процесу та спостереження, що обидва вважаються шумами з багатовимірним нормальним розподілом з нульовим середнім значенням з коваріаціями Qk та Rk відповідно. uk є вектором керування.

Функція f може використовуватися для обчислення передбачуваного стану з попередньої оцінки, і, аналогічно, функція h може використовуватися для обчислення передбачуваного вимірювання з передбаченого стану. Проте, f та h не можуть застосовуватися до коваріації безпосередньо. Натомість обчислюється матриця часткових похідних (матриця Якобі).

На кожному такті матриця Якобі обчислюється для поточних передбачених станів. Ці матриці можуть використовуватися у рівняннях фільтру Калмана. Цей процес, по суті, лінеаризує нелінійну функцію навколо поточної оцінки.

Наведена вище рекурсія є розширеним фільтром Калмана першого порядку. Розширені фільтри Калмана вищих порядків може бути отримано збереженням більшої кількості членів розкладу в ряд Тейлора. Однак, розширеним фільтрам Калмана вищих порядків властиво забезпечувати перевагу в продуктивності лише за малого шуму вимірювання.

На відміну від свого лінійного двійника, розширений фільтр Калмана не є оптимальним оцінювачем (звісно, він є оптимальним, якщо моделі як вимірювання, так і переходу стану є лінійними, і в цьому випадку розширений фільтр Калмана є ідентичним звичайному). На додачу, якщо початкова оцінка стану є невірною, або якщо процес змодельовано некоректно, фільтр може швидко розходитися внаслідок своєї лінеаризації. Іншою проблемою розширеного фільтру Калмана є те, що оціненій матриці коваріації властиво недооцінювати справжню матрицю коваріації, і відтак призводити до ризику

неузгодженості у статистичному сенсі без додавання «стабілізувального шуму».

При всьому цьому, розширений фільтр Калмана може пропонувати прийнятну продуктивність, і вірогідно є стандартом де-факто у навігаційних системах та GPS. Розширений фільтр Калмана створюється лінеаризацією моделі сигналу навколо оцінки поточного стану та використанням лінійного фільтру Калмана для передбачення наступної оцінки. Це є намаганням зробити локальний оптимальний фільтр, однак, воно не є обов'язково стабільним, оскільки розв'язки базового рівняння Ріккаті не є гарантовано додатньовизначеними. Одним зі способів покращення продуктивності є штучний алгебраїчний метод Ріккаті, що покращує стабільність ціною оптимальності. Зберігається звична структура розширеного фільтру Калмана, але стабільність досягається вибором додатньовизначеного розв'язку штучного алгебраїчного рівняння Ріккаті для моделі функції передавального коефіцієнту.

Іншим шляхом покращення продуктивності розширеного фільтру Калмана є застосування результатів H-нескінченності з робастного керування. Робастні фільтри отримуються додаванням додатньовизначеного члену до рівняння Ріккаті. Цей додатковий член параметризується скаляром, що розробник може налаштовувати для отримання компромісу між двома критеріями продуктивності, середньоквадратичною помилкою, та піковою помилкою.

Нелінійним фільтром Калмана, що обіцяє поліпшення відносно розширеного фільтру Калмана, є беззапаховий фільтр Калмана (англ. unscented Kalman filter, UKF). У беззапаховому фільтрі Калмана щільність ймовірності наближується детермінованою вибіркою точок, що представляють базовий розподіл як нормальний. Нелінійне перетворення цих точок призначене бути оцінкою апостеріорного розподілу, моменти якого потім можна буде вивести з перетвореної вибірки. Це перетворення також відоме як беззапахове перетворення. Беззапаховому фільтрові Калмана властиво бути надійнішим (робастнішим) та точнішим за розширений фільтр Калмана у його оцінці помилки в усіх напрямках.

Розширений фільтр Калмана є чи не найуживанішим алгоритмом оцінювання для нелінійних систем. Однак, понад 35 років досвіду в спільноті оцінювання показали, що він є складним для реалізації, складним для налаштування, і є надійним лише для систем, що є майже лінійними на масштабі часових проміжків уточнень. Багато з цих складнощів виникають з причини використання ним лінеаризації.

Інваріантний розширений фільтр Калмана є видозміненою версією розширеного фільтру Калмана для нелінійних систем, що володіють симетріями (або інваріантностями). Він об'єднує переваги як розширеного фільтру Калмана, так і нещодавно представлених фільтрів зі збереженням симетрії. Дійсно, замість використання лінійного члена уточнення на базі лінійної помилки виходу він використовує геометрично пристосований член уточнення на базі інваріантної помилки виходу; аналогічно, матриця передавального коефіцієнту уточнюється не з лінійної, а з інваріантної помилки стану. Головною вигодою є те, що рівняння передавального коефіцієнту та коваріації сходяться до сталих значень на значно більшій множині траєкторій, ніж точки рівноваги, як це є у випадку розширеного фільтру Калмана, що призводить до кращого сходження оцінювання.

Коли моделі переходу стану та спостереження, — тобто, функції передбачення та уточнення та h, — є сильно нелінійними, розширений фільтр Калмана може демонструвати особливо погану роботу. Це відбувається тому, що коваріація передається через лінеаризацію базової нелінійної моделі. Беззапаховий фільтр Калмана використовує метод детерміністичної вибірки, відомий як беззапахове перетворення, для вибору мінімального набору опорних точок (що називаються сигма-точками) навколо середнього значення. Ці сигма-точки (англ. sigma points) потім пропускаються через нелінійні функції, з чого отримуються середнє значення та коваріація оцінки. Результатом є фільтр, що точніше захоплює справжнє середнє значення та коваріацію. (Це можна перевіряти за допомогою вибірки Монте-Карло, або розкладу апостеріорної статистики в ряд Тейлора.) На додачу, цей метод усуває вимогу явного

обчислення матриць Якобі, що для складних функцій може бути саме по собі складною задачею (наприклад, вимагаючи складних похідних при аналітичній реалізації, або будучи обчислювально витратним при реалізації чисельній).

Як і з розширеним фільтром Калмана, передбачення беззапахового фільтру Калмана може використовуватися незалежно від уточнення беззапахового фільтру Калмана, у комбінації з лінійним (або навіть розширеним) уточненням, або навпаки.

Оцінюваний стан та коваріація поповнюються середнім значенням та коваріацією шуму процесу.

2.3 Метод Монте-Карло

Ме́тод Мо́нте-Ка́рло (за назвою міста Монте-Карло, Монако, яке відоме своїми казино) — загальна назва групи числових методів, основаних на одержанні великої кількості реалізацій стохастичного (випадкового) процесу, який формується у той спосіб, щоб його ймовірнісні характеристики збігалися з аналогічними величинами задачі, яку потрібно розв'язати. Використовується для розв'язування задач у фізиці, математиці, економіці, оптимізації, теорії управління тощо.

Метод Монте-Карло — це метод імітації для приблизного відтворення реальних явищ. Він об'єднує аналіз чутливості (сприйнятливості) і аналіз розподілу ймовірностей вхідних змінних. Цей метод дає змогу побудувати модель, мінімізуючи дані, а також максимізувати значення даних, які використовуються в моделі. Побудова моделі починається з визначення функціональних залежностей у реальній системі. Після чого можна одержати кількісний розв'язок, використовуючи теорію ймовірності й таблиці випадкових чисел.

Метод Монте-Карло широко використовується у всіх випадках симуляції

на ЕОМ.

Не існує єдиного методу Монте-Карло, цей термін описує великий і широко використовуваний клас підходів. Проте ці підходи використовують в своїй основі єдиний шаблон:

– Визначити область можливих вхідних даних.

– Випадковим чином згенерувати вхідні дані із визначеної вище області за допомогою деякого заданого розподілу ймовірностей.

– Виконати детерміновані обчислення над вхідними даними.

– Проміжні результати окремих розрахунків звести у кінцевий результат.

$$P\left(\left|\frac{\rho_N}{N} - m\right| \leq \left(\Phi^{-1}\left((1+\alpha)/2\right)\right)\frac{b}{\sqrt{N}}\right) \approx \alpha$$

Рівномірно розкидати деякі об'єкти однакового розміру по всій площі квадрата. Наприклад, це можуть бути зерна рису.

Оскільки дві області знаходяться в співвідношенні рі /4, об'єкти повинні потрапити в області приблизно в тій же пропорції. Таким чином, підрахувавши кількість об'єктів в колі і розділити на загальну кількість об'єктів в квадраті, отримаємо наближене значення рі /4.

Помноживши результат на 4 буде отримано наближене значення власне самого рі .

$$M\zeta = \int_a^b [g(x)/p_\xi(x)]p_\xi(x)dx = I$$

Енріко Фермі в 1930-х і Станіславу Уламу в 1946 році першим прийшла в голову ідея подібного методу. Улам пізніше зв'язався з Джоном фон Нейманом, щоб працювати над ним.

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ АЛГОРИТМІВ НАВІГАЦІЇ

3.1 Узагальнена структура програмно-апаратної системи

Для моделювання динамічних дискретних систем мною обране програмне середовище PetriNet. Мережа Петрі представляє собою орієнтований дводольний граф який має вершити двох типів – позиції та переходи.

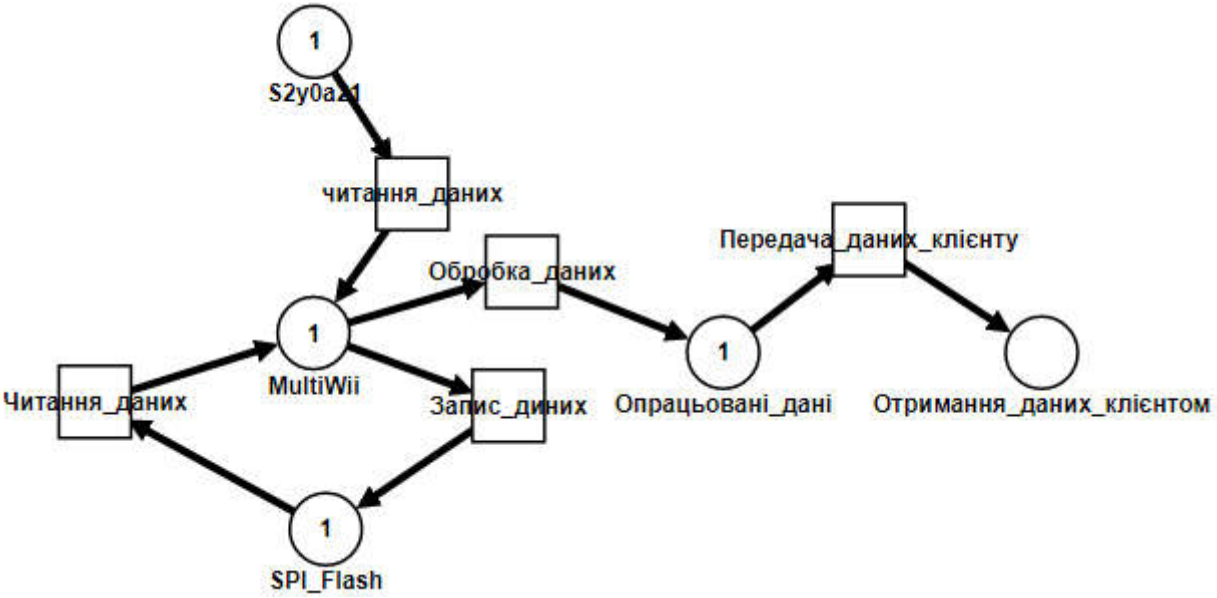

Рисунок 3.1 – Структурна схема модуля системи пристрою визначення відстані на основі Sharp 2y0a21 на основі мережі Петрі.

Отримані результати дають змогу стверджувати, що мережа є живою, усі стани досяжні та відсутні підвисання.

Базова структура програми для контролера досить проста та складається з 2х частин: завантажувача та виконуваної програми розробника.

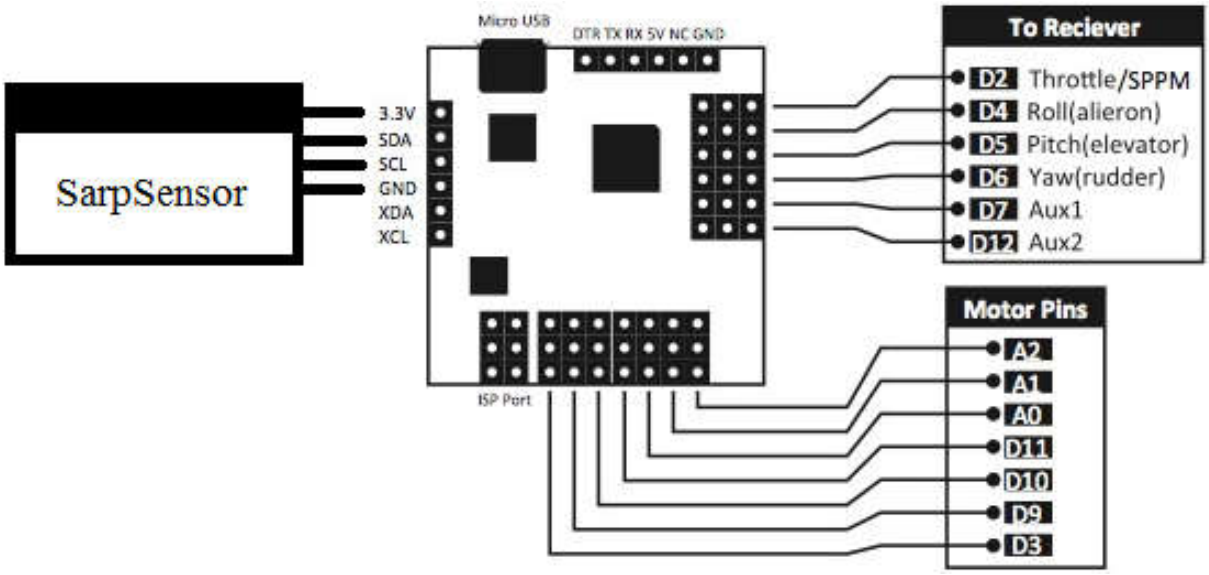

Рисунок 3.2 – Схема підключення контролерів та переферії.

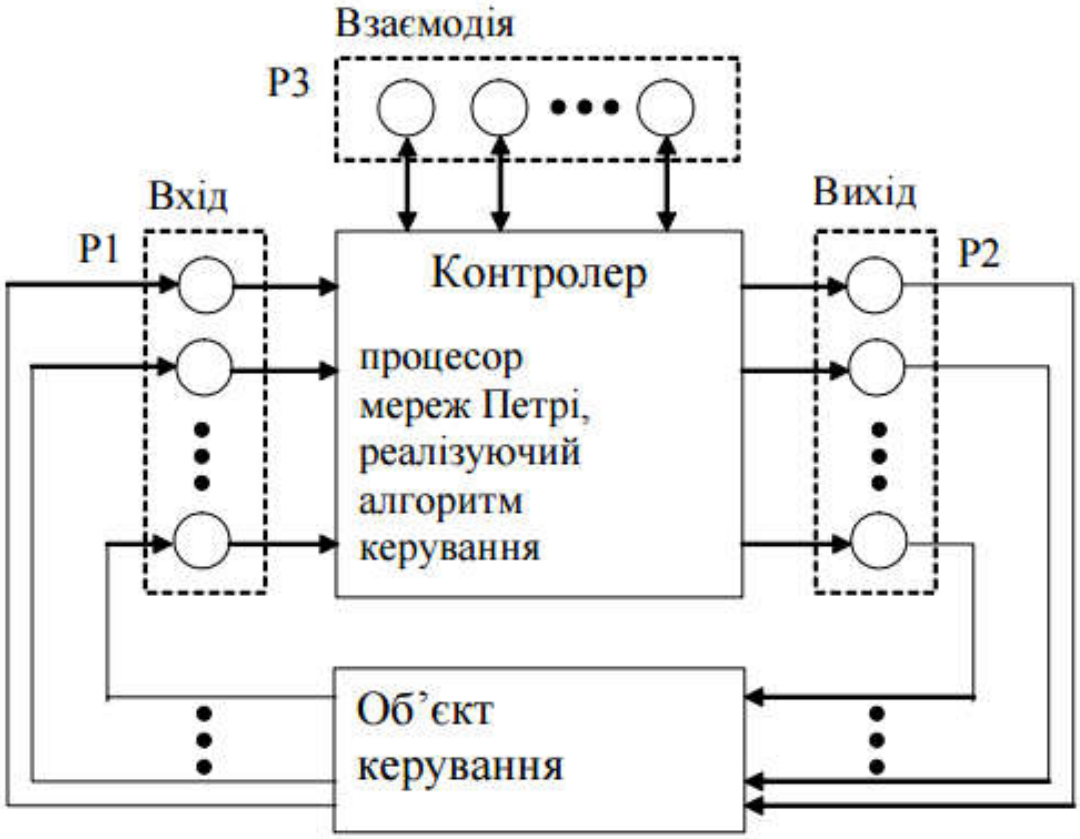

Рисунок 3.3 – Взаємодія контролера з системою керування.

Для розробки прогами під контролер MultiWii було використано AtmelStudio 7 та модуль VisualMicro (рисунок 3.4-3.5)

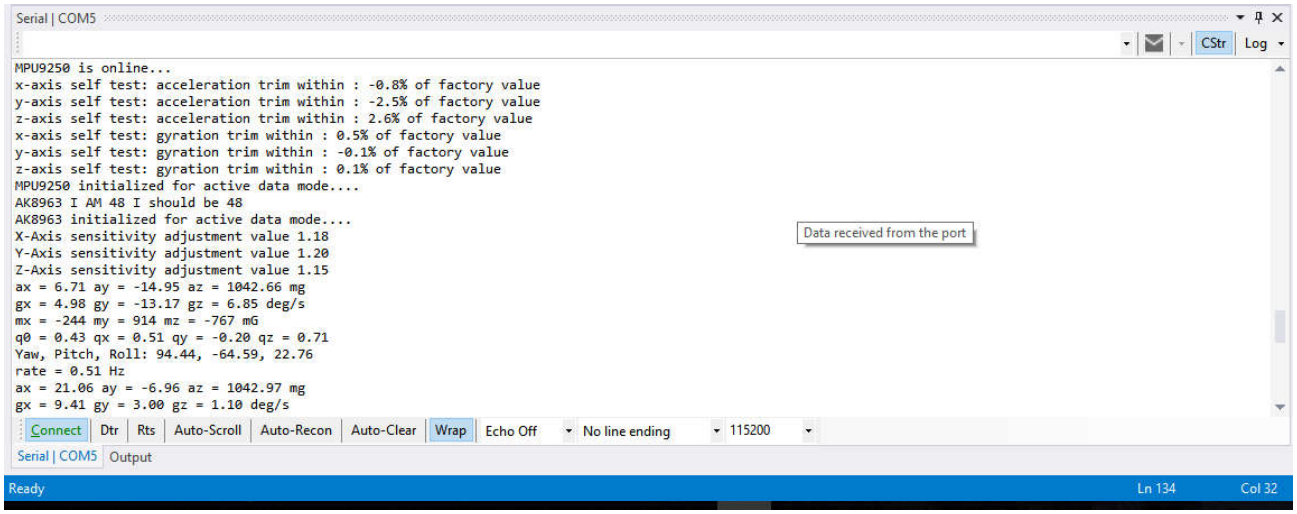

Рисунок 3.4 - AtmelStudio 7, Ініціалізація обміну даними контролера MultiWii та підключеною переферією.

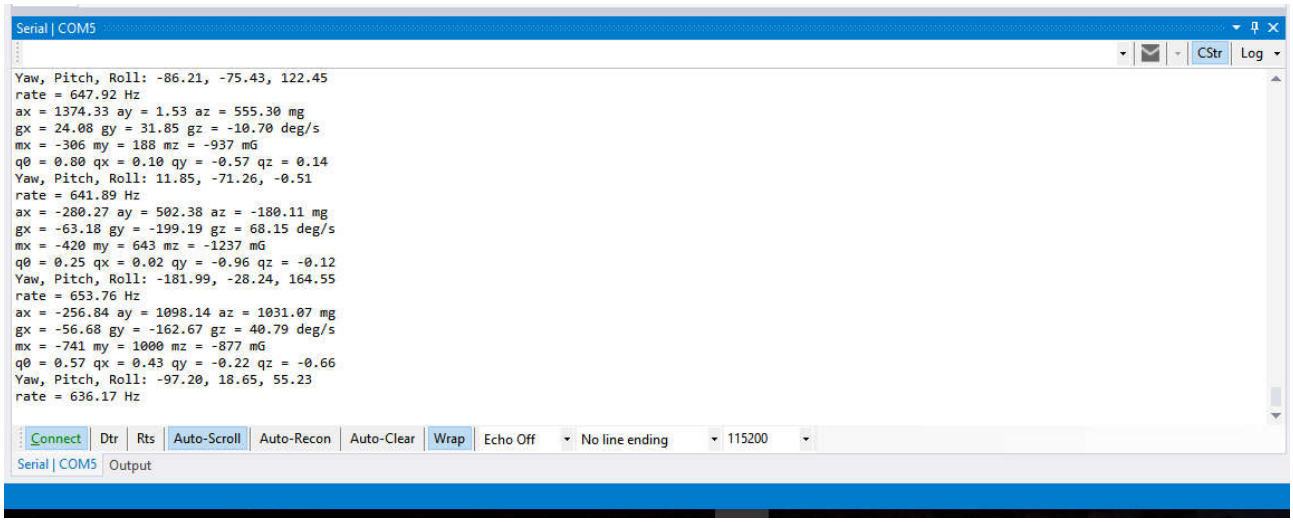

Рисунок 3.5 - AtmelStudio 7, Результат виконання програми – показники датчиків в human readable форматі.

Програмна система управління складається з таких блоків (ДП.КСМ.07024/09.00.00.001.С1):

- модуль збору даних з оптичних давачів;
- модуль взаємозв’язку із оптичним сенсором;
- модуль керування двигунами;
- модуль синхронізації;
- модуль маневрів квадрокоптера та управління двигунами коліс;

– модуль побудови карти руху;

– модуль визначення напрямку руху.

Модуль збору даних з оптичних давачів. Він відповідає за отримання даних від оптичного сенсора та подання інформації про перешкоди до контролера польоту.

Модуль взаємозв’язку із оптичним сенсором призначений взаємодію мікроконтролера із оптичним сенсором. Основною задачею даного модуля є передача даних про відстань до перешкод отриманих від оптичного сенсора до контролера польоту.

Для роботи з сенсором, уже є проста бібліотека - SharpIR. Приклад використання, з видачею відстані до об'єкту в послідовний порт - SharpIR подано на рисунку 3.6.

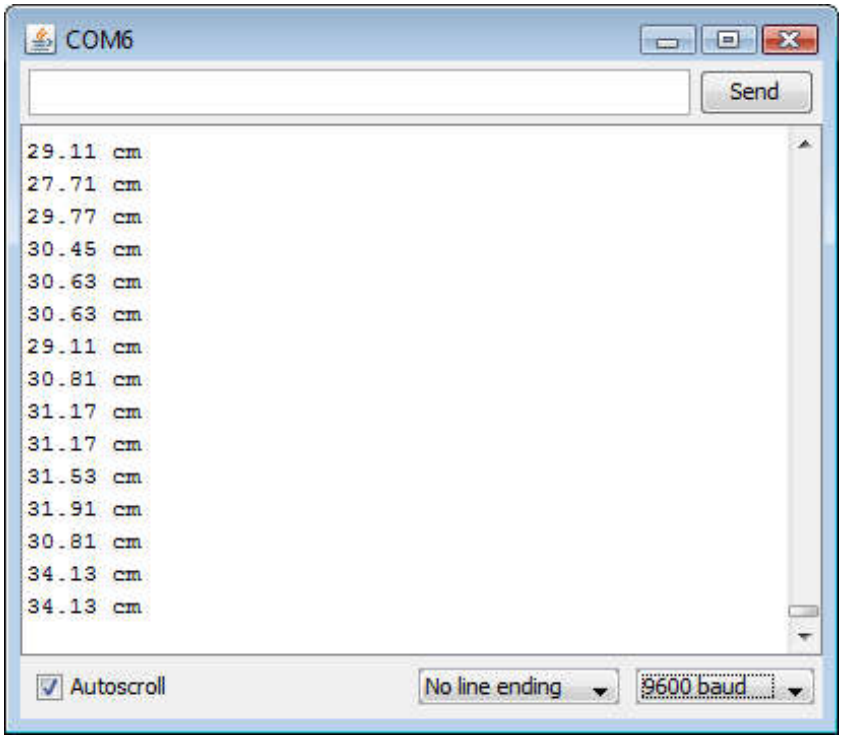

Рисунок 3.6 – Визначення відстані до обєкту за допомогою дальномірів.

Модуль взаємозв’язку із оптичним сенсором включає в себе інтерфейс підключення оптичного сенсора Sharp до мікроконтролерної платформи.

Він складається з трьох виводів:

– Плюс живлення (Vcc).
– Вхід Vo.
– Нуль живлення (GND).

На входи живлення подається постійна напруга 5 В. Датчик споживає в робочому режимі 30 мА.

Вхід Vo підключається до будь-якого аналогового виводу мікроконтролера. На цему вході читаємо аналоговий сигнал який приходить з оптичного сенсора. Після прийому відбитого сигналу, сенсор формує на виході Vo аналоговий сигнал, значення якого пропорційне відстані до перешкоди. (рисунок 3.7).

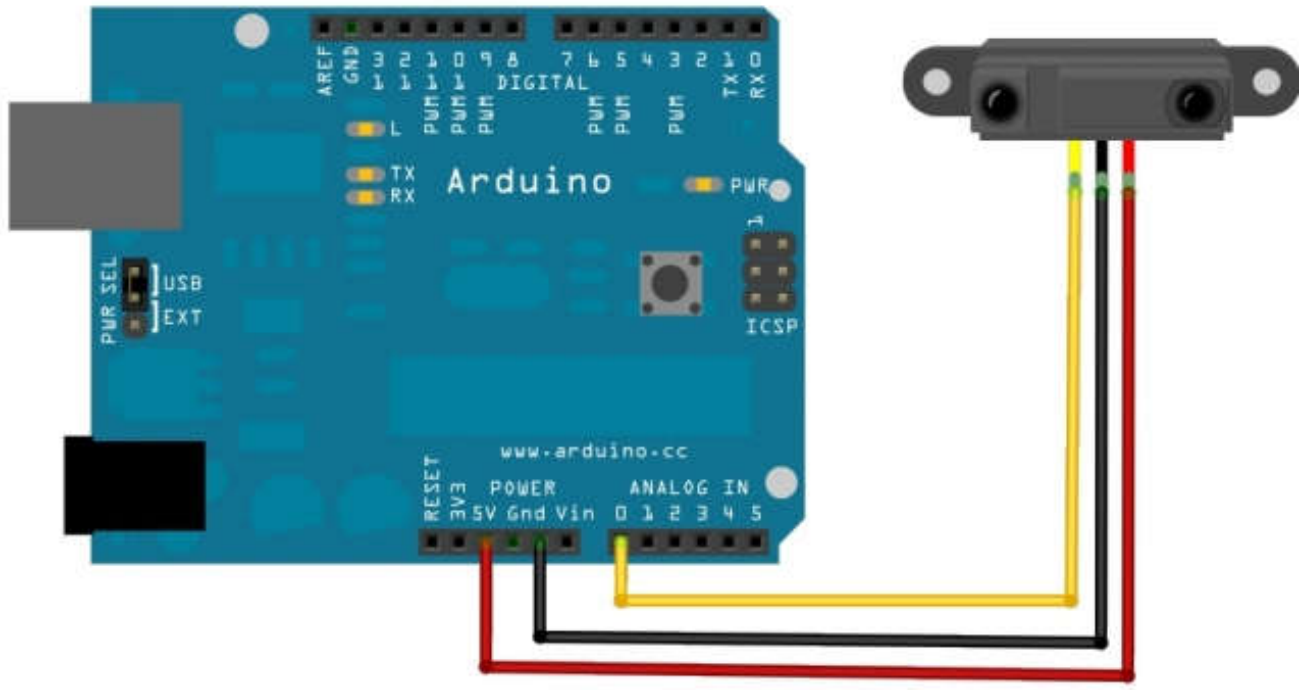

Рисунок 3.7 - Схема підключення оптичного сенсора Sharp 2y0a21.

.

Модуль управління двигуном призначений за керування швидкістю обертів двигуна за допомогою широтно-імпульсної модуляції (ШІМ). Чим довший коефіцієнт, тим з більшою швидкістю обертатиметься двигун (рисунок 3.6).

Основною задачею даного модуля є :

- збурення вільного руху об'єкта двигуном тільки за наявності керуючого сигналу;
- пропорційна залежність між обертовим моментом двигуна та керуючим сигналом комп'ютера (контролера).

Програмне керування двигуном представлено на рисунку 3.8.

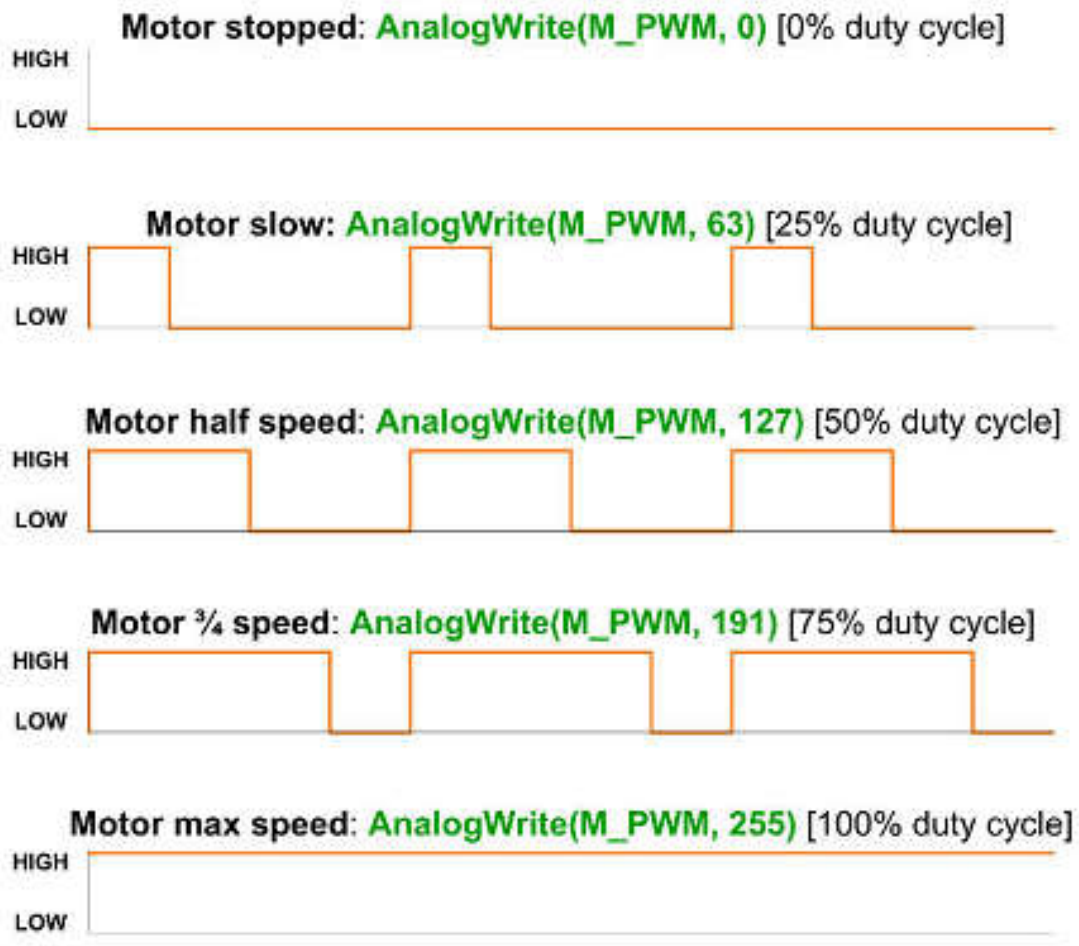

Рисунок 3.8 – Управління швидкістю обертів двигуна за допомогою ШІМ.

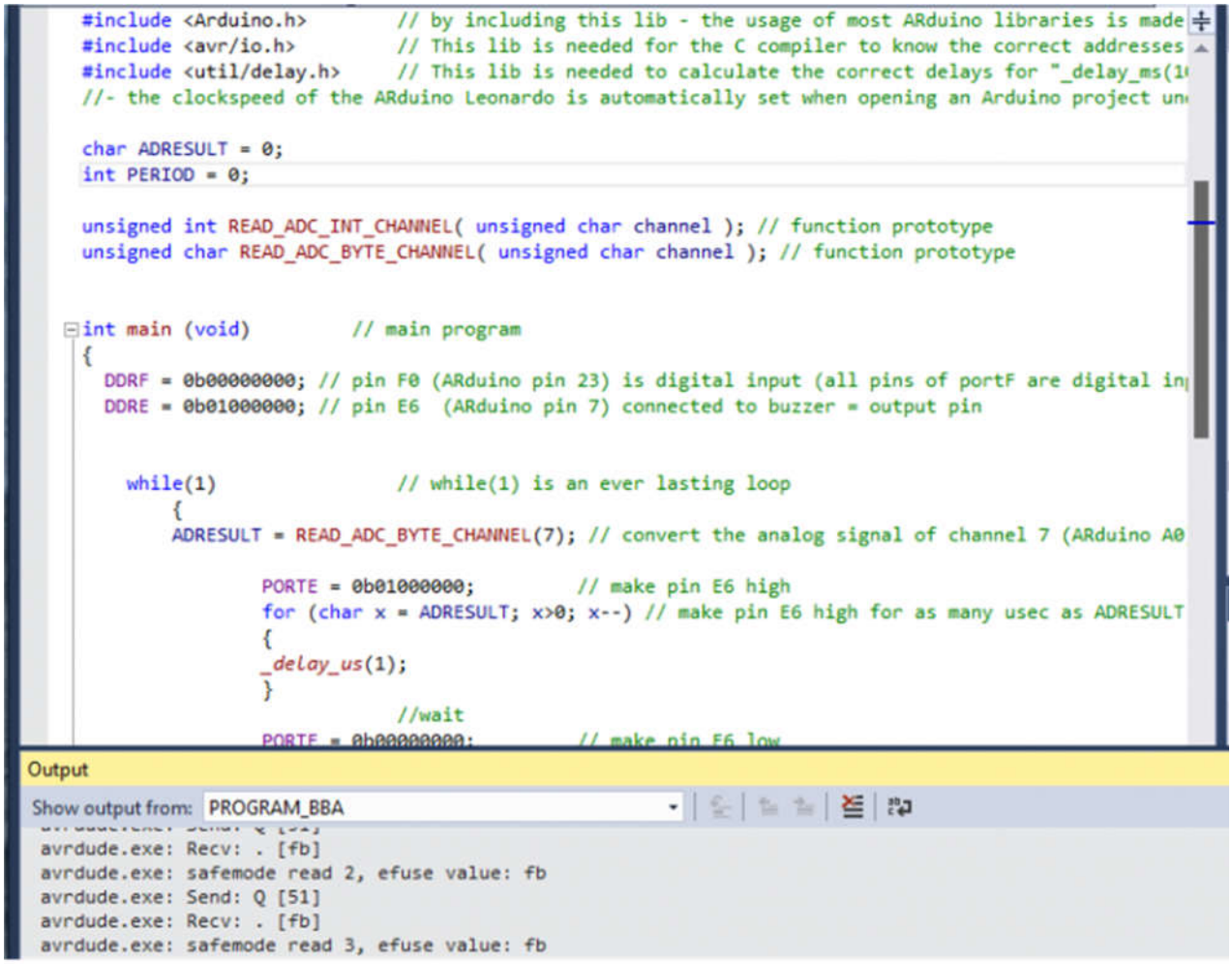

Рисунок 3.9 – Програмне керування двигуном

Модуль синхронізації відповідає за злагоджену роботу усіх блоків управління роботом.

Модуль маневрів робота та управління двигунами відповідає за синхронізовану роботу усіх гвинтів при польоті.

Модуль побудови карти руху робота відповідає складання карти середовища, локалізацію (визначення положення квадрокоптера на мапі за його координатами в середовищі) і планування маршруту (вибір оптимального шляху, що веде до мети). Перші два етапи є взаємо-пов'язаними (карта середовища потрібна для оцінки власного становища робота, знання якої, в свою чергу, необхідно для побудови карти).

Модуль визначення напрямку руху. Одним із головних параметрів при цьому є визначення напрямку обходу перешкод на площині і в просторі. Задача

даного модуля зводиться до моделювання навколишнього середовища та планування маршрутів (із врахуванням перешкод). Даний модуль є голоною складовою системи керування мобільним роботом.

Ціллю переміщення робота є кінцева точка, яка задається користувачем або маркується відповідним маркером (світловим, електромагнітним, звуковим) на карті місцевості.

Для організації руху квадрокоптера пропонується принцип, оснований на аналізі результатів вимірювань оптичних сенсорів відстаней від робота до перешкоди, як в статичному положенні, так і в процесі руху робота. Наявність перешкоди в напрямку руху робота визначається за часом приходу відбитого імпульсу, випроміненого і прийнятого сенсором в інфрачервоному діапазоні частот.

У процесі руху робота сенсор та вимірювальна система безперервно здійснюють вимірювання відстані до перешкоди. При досягненні мінімально допустимого порогового значення (задається опціонально) від квадрокоптера до перешкоди. У разі якщо відстань виявляється більше, робот рухається в напрямку найбільшого запасу руху (наліво). За відсутності безперешкодних напрямків руху здійснюється зупинка робота і рух у зворотний бік.

Управління локальними переміщеннями відомим маршрутом здійснюється на підставі інформації про характер перешкод на шляху робота.

Якщо визначити маршрут руху як послідовність опорних пунктів (підцілей) руху, що включає початкове і кінцеве (цільове) положення робота, то завдання побудови маршруту включає формування безлічі підцілей і подальший вибір такої її підмножини, яка оптимізує рух робота.

Процесу побудови маршруту руху робота передує складання карти середовища. Спочатку формується карта робочої зони робота, при цьому зовнішнє середовище дискретизується, і кожній ділянці, що містить перешкоду, ставиться у відповідність інформація про тип цієї перешкоди. Побудова карти відбувається одночасно з дослідженням зовнішнього середовища.

Нехай в початковий момент часу зовнішнє середовище не досліджене, а робот знаходиться в центрі вільної ділянки. Якщо положення сенсорів у позиції 0^0, тоді виконується цикл, тобто йде сканування середовища, визначення відстані до перешкоди і занесення значень у карту середовища. Наступним кроком є переміщення робота та продовження виконання циклу.

Виходячи з цього, будуємо алгоритм складання карти місцевості.

Після складання карти середовища робот повинен використовувати її в процесі переміщень до цілі. Однак отримана карта не може бути абсолютно точною через похибки вимірювань. Тому робот після кожного переміщення повинен виконувати уточнення карти, позиції на ній. Складання карти місцевості у вигляді алгоритму наведено в ДП.КСМ.07024/09.00.00.002.С1.

Карта перешкод у системі координат відображає відстань до перешкод відносно положення робота.

Алгоритм функціонує за таким принципом:

1. Активація контролерів: оптичних, польоту, двигунів,.
2. Відслідковування положення платформи у середовищі.
3. В тілі циклу відбувається сканування середовища.
4. На наступному етапі відбувається визначення відстані до перешкоди і занесення даних, отриманих від оптичних сенсорів, в карту середовища.
5. Наступним кроком є керування двигунами відповідно до отриманих даних від оптичних давачів та карти середовища.
6. Визначення відстані до перешкоди і занесення даних, отриманих від сенсорів, в карту середовища.
7. Формування карти середовища для руху робота.

У результаті цього формується карта де записані координати перешкод на шляху робота. При цьому його завданням є знаходження оптимального шляху руху до цілі. Діючи відповідно до описаного алгоритмом навігації по карті, робот досягає мети по траєкторії, яка є найбільш оптимальною. З огляду на те що локальна модель навколишнього середовища отримана з деякою

часткою абстракції, в процесі роботи може виникнути невизначеність у визначенні положення якого-небудь з об'єктів. Це може бути обумовлено, наприклад, нерівномірністю розташування сенсорів по периметру, в результаті чого отримана інформація може бути неповною.

Одна з основних проблем, що виникають при переміщенні робота у середовищі це необхідність виконання відповідних маневрів з метою уникнення зіткнень [15]. Для плавного маневрування необхідно враховувати перешкоди як безпосередньо на шляху руху, так і в напрямку ймовірного маневру, що вимагає внесення коректив у дані, отримані навігаційною системою. Подібні ситуації часто виникають при русі у вузьких коридорах, офісних приміщеннях і виставкових залах, скрізь, де переміщаються люди. Тому сенсорна система повинна враховувати дані про стан у просторі, а також постійно локалізувати в ньому об'єкти, що перешкоджають руху.

Для обчислення напрямку необхідного маневру система оптичної навігації повинна послідовно реєструвати положення оточуючих об'єктів в різні моменти часу, змінюючи тим самим профіль зони безпеки. Для простоти представлення зони безпеки об'єкти розглядаються у вигляді геометричних примітивів циліндричної форми. Щоб уникнути зіткнення з об'єктом, що перешкоджає руху, в процесі планування маршруту руху планується необхідний маневр. Напрямок маневру визначається з числа можливих.

Саме завдання планування маневру з обходом перешкоди може бути легко вирішена одним з відомих методів [16].

Таким чином, комплексування інформації від різних модулів сенсорної системи дозволяє оперативно реагувати на небезпеку від динамічних об'єктів поза сектором огляду оптичного сенсора. Крім того, застосування системи оптичної навігації, дозволяє підвищити точність визначення положення.

Додаткова інформації про довкілля показів інфрачервоних сенсорів і побудова локальної моделі карти середовища, дозволяє роботу функціонувати в таких середовищах, де застосування ультразвукових сканерів виявляється марним, наприклад, у приміщеннях з великим числом звукопоглинаючих

поверхонь. Використання інформації про наявність перешкод у зоні безпеки дає можливість системі управління своєчасно вибрати напрямок для маневру, в ході виконання якого вдасться уникнути зіткнення. Застосування цього підходу до розробки сенсорної системи дозволяє поліпшити процес пересування в середовищі, а також розширити область застосування подібних мобільних роботів.

Виходячи з цього, застосування описаного методу дозволяє роботу будувати карту зовнішнього середовища й ефективно використовувати її для переміщення в ній [17].

Після включення живлення перевіряється наявність сигналу від оптичних сенсорів. Якщо сигнал від оптичних сенсорів відсутній, на двигуни не подаються команди. У разі, якщо наявність сигналу виявлена, виконується порівняння відстаней до перешкоди. У разі рівності отриманих відстаней і за умови, що вони не менше критичного значення, робот рухається вперед. При неспівпаданні відстаней, отриманих з сенсорів, виконується поворот робота. Якщо отримані відстані менші за критичні, на двигуни робота подається команда, що забезпечує поворот на 90 градусів.

Управління рухом робота здійснюється за наступним алгоритмом: якщо відстань по осі робота до об'єкту, визначене за допомогою інфорачервоного сенсора, складає більше мінімально критичного значення, робот рухається вперед, якщо відстань менша - робот повертає за рахунок обертання гвинтів з певною частотою, при цьому відстань до об'єкту визначається постійно. Використовуваний алгоритм приведений в ДП.КСМ.07024/09.00.00.003.С1.

Описаний вище алгоритм виявлення перешкод не дозволяє врахувати перешкоди, розташовані в сліпій зоні робота, тому можливі його зіткнення з перешкодами, що не потрапляють у зону видимості оптичних сенсорів.

Для унеможливлення зіткнень необхідно отримувати дані про увесь простір перед роботом. Фізично це важко здійснити - в першу чергу потрібна велика кількість сенсорів. Навіть якщо припустити можливість їх установки, для обробки отримуваних від них даних потрібно багато ресурсів: більше

оперативної пам'яті, більше частота процесора, що неприйнятно для безпілотників де потрібно (в ідеальних умовах) робити обчислення в реальному часі, оскільки це веде до ускладнення і здорожчання. Тому необхідно використовувати мінімальну кількість сенсорів, розташовуючи їх так, щоб вони давали максимально необхідну кількість інформації для уникнення зіткнень [18].

На цьому етапі використовувався один оптичний сенсор, але він встановлений на максимальній висоті, яку можуть досягати окремі частини робота.

Оскільки необхідно по максимуму отримувати інформацію про перешкоди, розташовані на шляху робота, необхідно обробляти дані з усього кута дії сенсора, і враховувати відстані до перешкод, отримувані на усій області дії дільноміра [19].

Також розроблено контролер попередньої обробки сигналу від інфрачерфоних дальномірів Sharp 2y0a21 (рисунок 3.10). Даний контролел здійснюватиме попередню обробку зашумленого аналогового сигналу фільтром Калмана, та передаватиме польотному контролеру MultiWii оброблені дані у цифровому вигляді.

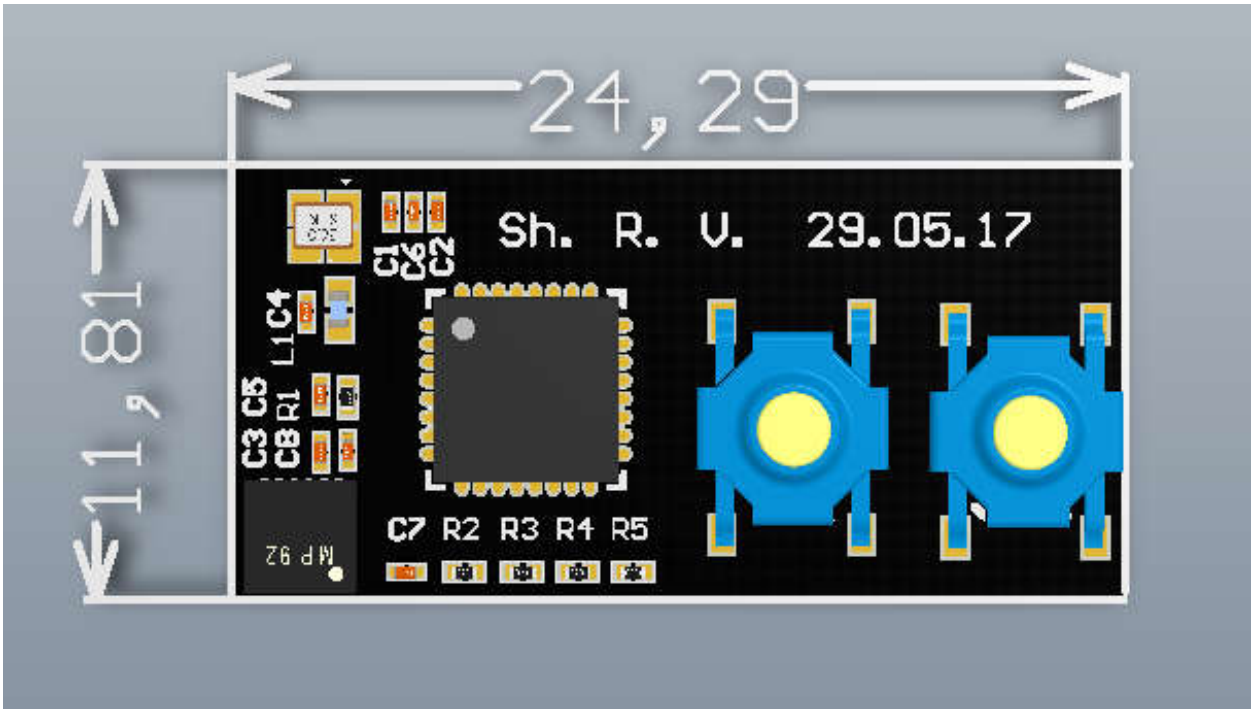

Рисунок 3.10 – Контролер попередньої обробки аналогового сигналу.

3.2 Реалізація алгоритму SLAM

При вирішенні задач навігації роботів, одночасна локалізація і картографування (англ. simultaneous localization and mapping - SLAM) є алгоритмічною обчислювальною задачею побудови і оновлення мапи невідомого оточення з одночасним відстежуванням місцезнаходження рухаючись по ньому (рисунок 3.11). Популярні методи вирішення включають в себе Фільтр часток і Розширений фільтр Калмана.

Алгоритми SLAM обмежуються наявними ресурсами, таким чином не можуть бути абсолютно досконалим. Опубліковані методи і підходи реалізовані в безпілотних автомобілях, безпілотних літаючих засобах, автономних підводних апаратах, планетоходах, згодом виникли в побутових роботах і навіть всередині людського тіла.

Дана послідовність даних спостереження сенсора за дискретні проміжки часу задачею SLAM є розрахувати і визначити розташування та мапу оточення. Всі величини зазвичай ймовірнісні, тому необхідно обчислити:

$$P = (m_t, x_t | o_{1:t})$$

Застосування правила Баєса дає основу для послідовного оновлення апостеріорного розташування, при даній мапі і функції переходу

$$P = (x_t | x_{t-1})$$

$$P(x_t | o_{1:t}, m_t) = \sum_{m_{t-1}} P(o_t | x_t, m_t) \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1} | m_t, o_{1:t-1}) / Z$$

Аналогічно мапа може оновлюватися послідовно наступним шляхом

$$P(m_t | x_t, o_{1:t}) = \sum_{x_t} \sum_{m_t} P(m_t | x_t, m_{t-1}, o_t) \, P(m_{t-1}, x_t | o_{1:t-1}, m_{t-1})$$

Побудова топологічних мап це метод представлення довколишнього світу, який узагальнює структуру (тобто, топологію) оточення замість

створення геометрично точної мапи. Топологічні методи SLAM використовуються для підвищення загальної узгодженості метричних алгоритмів SLAM.

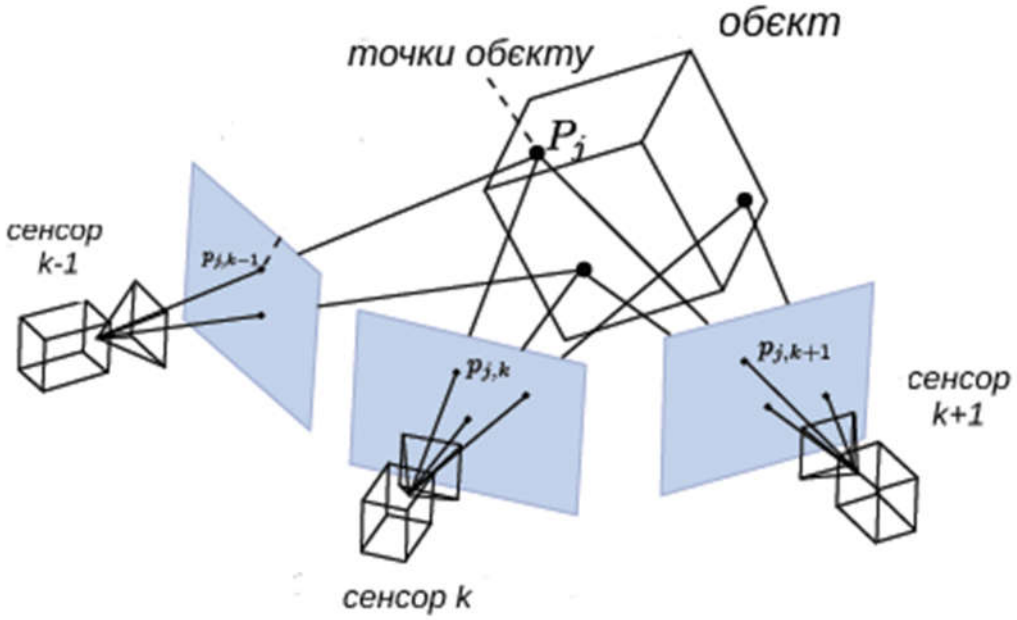

Рисунок 3.11 – Локалізація обєкту.

На противагу тому, для топологічного представлення світу існують мапи у вигляді поверхонь, які використовують масиви оцифрованих елементів (зазвичай квадратних, або гексагональних клітин), і роблять припущення яка клітина зайнята в конкретний момент. Зазвичай клітини вважаються статистично незалежними, щоб спростити обчислення.

3.3 Реалізація розширеного фільтру Калмана

У теорії статистичного оцінювання розширений фільтр Калмана — це нелінійна версія фільтру Калмана, що лінеаризується навколо оцінки поточного середнього значення та коваріації. У випадку визначених моделей переходу розширений фільтр Калмана було визнано стандартом де-факто у теорії оцінювання нелінійних станів, навігаційних системах та GPS.

Фільтр Калмана є оптимальною оцінкою для моделей лінійних систем з додатковим незалежним білим шумом як у системі переходу, так і в системі

вимірювання. На жаль, більшість систем у техніці є нелінійними, тому одразу було зроблено певну спробу застосування цього методу фільтрування до нелінійних систем. Більшість цієї роботи було зроблено у дослідницькому центрі Еймса Наса. Розширений фільтр Калмана пристосував прийоми з числення, а саме багатовимірні розклади в ряди Тейлора, для лінеаризації моделі навколо робочої точки. Якщо модель системи (як описано нижче) не достатньо добре відома, або є неточною, то для оцінювання застосовуються методи Монте-Карло, зокрема частинкові фільтри. Методи Монте-Карло передують існуванню розширеного фільтру Калмана, але є обчислювально витратнішими для будь-якого простору станів з помірною кількістю вимірів.

У розширеному фільтрі Калмана моделі переходу стану та спостереження не повинні бути обов'язково лінійними функціями стану, натомість вони можуть бути нелінійними диференційовними функціями.

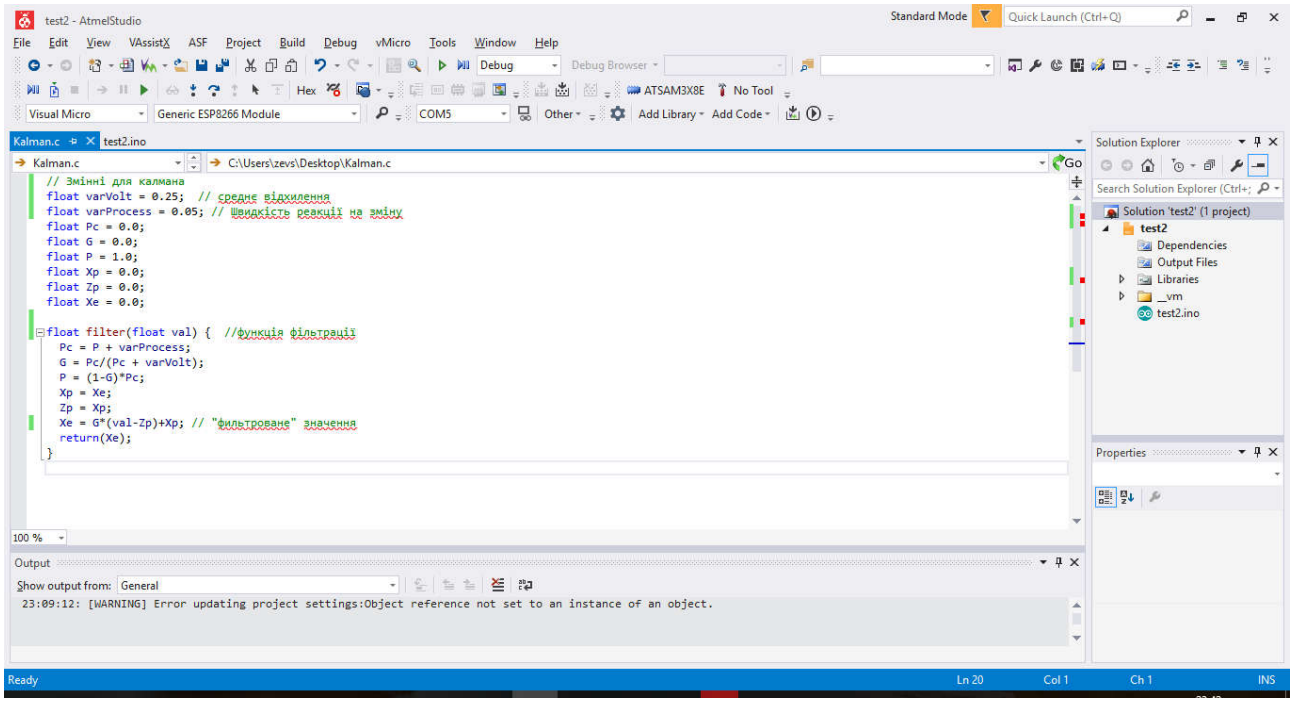

Рисунок 3.12 - AtmelStudio 7, Функція фільтру Калмана.

Розширені фільтри Калмана вищих порядків може бути отримано збереженням більшої кількості членів розкладу в ряд Тейлора. Однак,

розширеним фільтрам Калмана вищих порядків властиво забезпечувати перевагу в продуктивності лише за малого шуму вимірювання.

На відміну від свого лінійного двійника, розширений фільтр Калмана не є оптимальним оцінювачем (звісно, він є оптимальним, якщо моделі як вимірювання, так і переходу стану є лінійними, і в цьому випадку розширений фільтр Калмана є ідентичним звичайному). На додачу, якщо початкова оцінка стану є невірною, або якщо процес змодельовано некоректно, фільтр може швидко розходитися внаслідок своєї лінеаризації. Іншою проблемою розширеного фільтру Калмана є те, що оціненій матриці коваріації властиво недооцінювати справжню матрицю коваріації, і відтак призводити до ризику неузгодженості у статистичному сенсі без додавання «стабілізувального шуму».

При всьому цьому, розширений фільтр Калмана може пропонувати прийнятну продуктивність, і вірогідно є стандартом де-факто у навігаційних системах та GPS (рисунок 3.13). Розширений фільтр Калмана створюється лінеаризацією моделі сигналу навколо оцінки поточного стану та використанням лінійного фільтру Калмана для передбачення наступної оцінки. Це є намаганням зробити локальний оптимальний фільтр, однак, воно не є обов'язково стабільним, оскільки розв'язки базового рівняння Ріккаті не є гарантовано додатньовизначеними. Одним зі способів покращення продуктивності є штучний алгебраїчний метод Ріккаті, що покращує стабільність ціною оптимальності. Зберігається звична структура розширеного фільтру Калмана, але стабільність досягається вибором додатньовизначеного розв'язку штучного алгебраїчного рівняння Ріккаті для моделі функції передавального коефіцієнту.

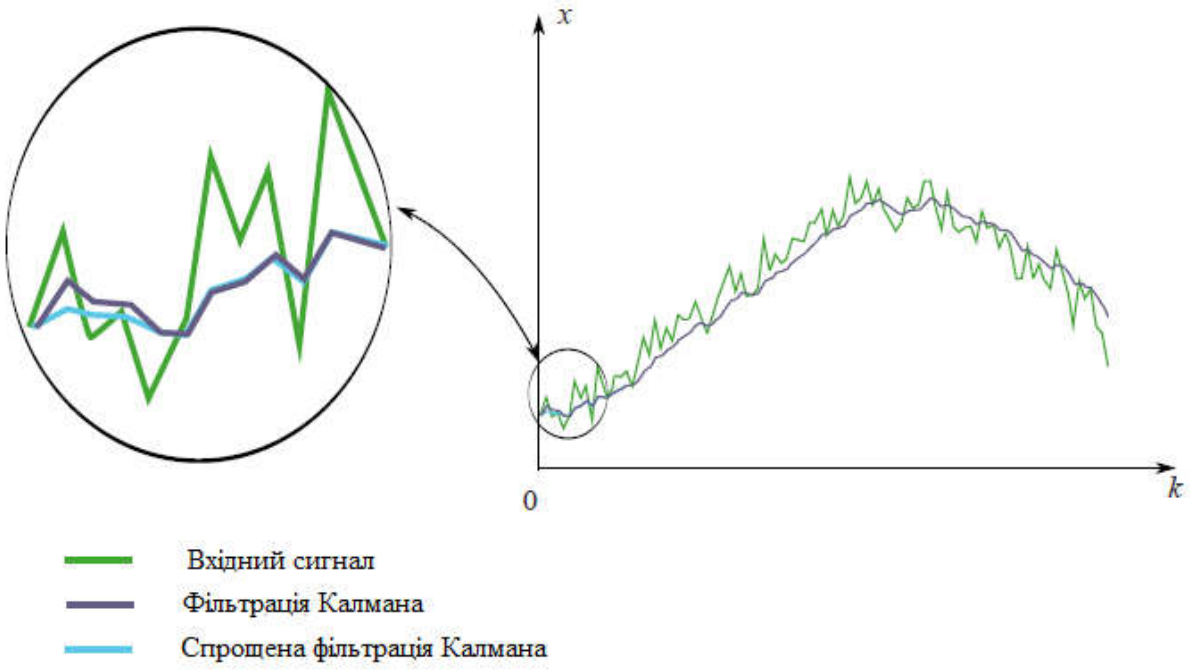

Рисунок 3.13 – Графік роботи фільтру Калмана.

Іншим шляхом покращення продуктивності розширеного фільтру Калмана є застосування результатів H-нескінченності з робастного керування. Робастні фільтри отримуються додаванням додатньовизначеного члену до рівняння Ріккаті. Цей додатковий член параметризується скаляром, що розробник може налаштовувати для отримання компромісу між двома критеріями продуктивності, середньоквадратичною помилкою, та піковою помилкою.

Нелінійним фільтром Калмана, що обіцяє поліпшення відносно розширеного фільтру Калмана, є беззапаховий фільтр Калмана (англ. unscented Kalman filter, UKF). У беззапаховому фільтрі Калмана щільність ймовірності наближується детермінованою вибіркою точок, що представляють базовий розподіл як нормальний. Нелінійне перетворення цих точок призначене бути оцінкою апостеріорного розподілу, моменти якого потім можна буде вивести з перетвореної вибірки. Це перетворення також відоме як беззапахове перетворення. Беззапаховому фільтрові Калмана властиво бути надійнішим (робастнішим) та точнішим за розширений фільтр Калмана у його оцінці

помилки в усіх напрямках.

Розширений фільтр Калмана є чи не найуживанішим алгоритмом оцінювання для нелінійних систем. Однак, понад 35 років досвіду в спільноті оцінювання показали, що він є складним для реалізації, складним для налаштування, і є надійним лише для систем, що є майже лінійними на масштабі часових проміжків уточнень. Багато з цих складнощів виникають з причини використання ним лінеаризації.

Передбачена оцінка стану та коваріація передбаченої оцінки:

$$x_{k|k-1} = f(x_{k-1|k-1}, u_{k-1})$$
$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1} + Q_{k-1}$$

Інваріантний розширений фільтр Калмана є видозміненою версією розширеного фільтру Калмана для нелінійних систем, що володіють симетріями (або інваріантностями). Він об'єднує переваги як розширеного фільтру Калмана, так і нещодавно представлених фільтрів зі збереженням симетрії. Дійсно, замість використання лінійного члена уточнення на базі лінійної помилки виходу він використовує геометрично пристосований член уточнення на базі інваріантної помилки виходу; аналогічно, матриця передавального коефіцієнту уточнюється не з лінійної, а з інваріантної помилки стану. Головною вигодою є те, що рівняння передавального коефіцієнту та коваріації сходяться до сталих значень на значно більшій множині траєкторій, ніж точки рівноваги, як це є у випадку розширеного фільтру Калмана, що призводить до кращого сходження оцінювання.

Коли моделі переходу стану та спостереження, — тобто, функції передбачення та уточнення та h, — є сильно нелінійними, розширений фільтр Калмана може демонструвати особливо погану роботу. Це відбувається тому, що коваріація передається через лінеаризацію базової нелінійної моделі. Беззапаховий фільтр Калмана використовує метод детерміністичної вибірки, відомий як беззапахове перетворення, для вибору мінімального набору опорних точок (що називаються сигма-точками) навколо середнього значення. Ці сигма-

точки (англ. sigma points) потім пропускаються через нелінійні функції, з чого отримуються середнє значення та коваріація оцінки. Результатом є фільтр, що точніше захоплює справжнє середнє значення та коваріацію. (Це можна перевіряти за допомогою вибірки Монте-Карло, або розкладу апостеріорної статистики в ряд Тейлора.) На додачу, цей метод усуває вимогу явного обчислення матриць Якобі, що для складних функцій може бути саме по собі складною задачею (наприклад, вимагаючи складних похідних при аналітичній реалізації, або будучи обчислювально витратним при реалізації чисельній).

Оцінюваний стан та коваріація поповнюються середнім значенням та коваріацією шуму процесу (рисунок 3.14).

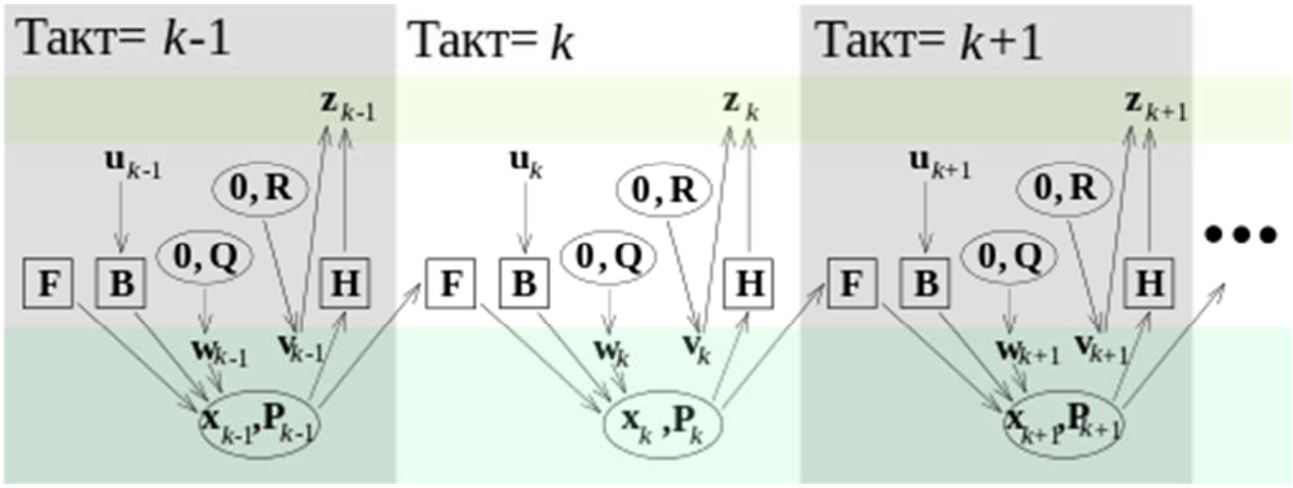

Рисунок 3.14 – Базова модель фільтру Калмана.

Квадрати представляють матриці. Еліпси представляють багатовимірні нормальні розподіли (із включеними середніми значеннями та матрицями коваріацій). Не обведені значення є векторами. У цьому простому випадку різні матриці є незмінними в часі, тому індекси пропущено, але фільтр Калмана дозволяє будь-якій з них змінюватися на кожному такті.

3.4 Тестування та реалізація роботи програмно-апаратного засобу

Після написання програми та її відлагодження контролер MultiWii проводить обробку даних які він отримує з контролера дальномірів Sharp. На основі цих даних польотний контролер має змогу коректувати траєкторію руху роботизованої платформи.

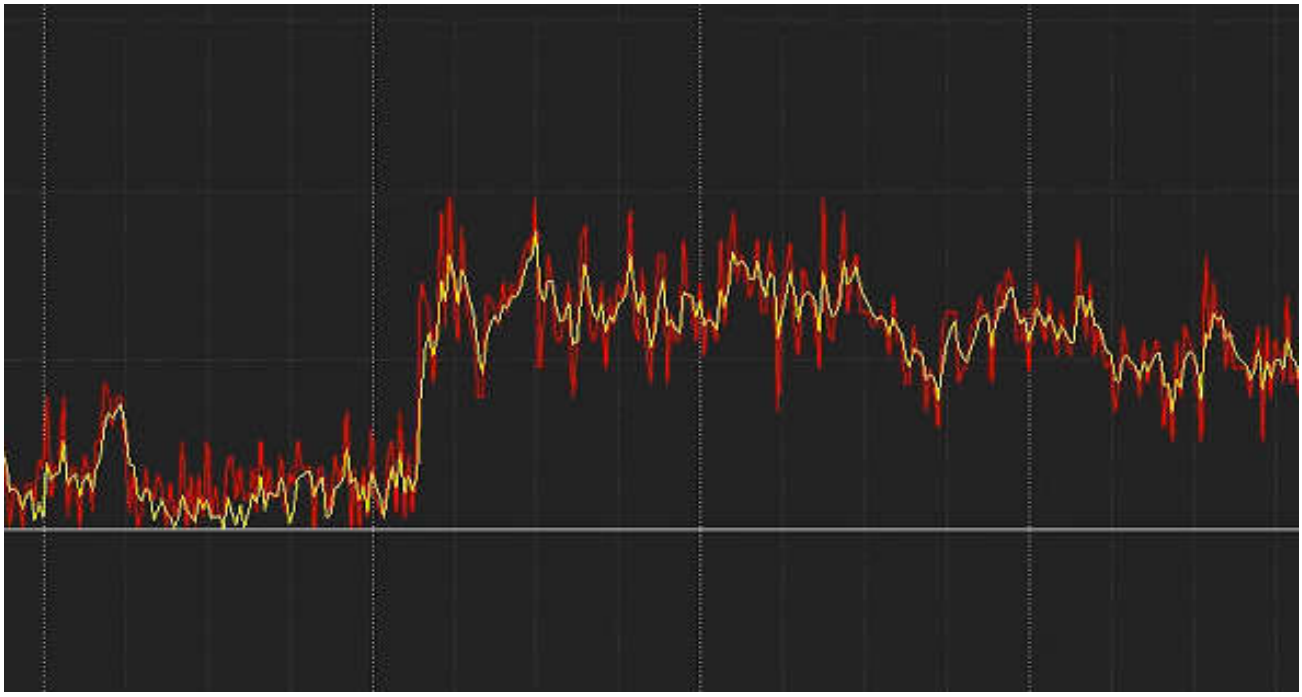

Рисунок 3.15 – Фільтрація зашумленого аналогового сигналу що надходить від дальноміра Sharp.

На рисунку 3.15 представлені графіки вхідного зашумленого аналогового сигналу від дальноміра Sharp та вихідного сигналу з контролера дальномірів.

Функція фільтрації:

```
float KalmanFilter(KalmanFilterTypeDef *Filter, float NewData) {
    Filter->Pc = Filter->P + Filter->varProcess;
    Filter->G = Filter->Pc / (Filter->Pc + Filter->varVolt);
    Filter->P = (1 - Filter->G) * Filter->Pc;
    Filter->Xp = Filter->Xe;
    Filter->Zp = Filter->Xp;
    Filter->Xe = Filter->G * (NewData - Filter->Zp) + Filter->Xp;
    return(Filter->Xe);          // фільтроване значення
}
```

Вихідний сигнал з контролера дальномірів позбавлений шумів які наявні на вхідному сигналі. За допомогою реалізації фільтра Калмана вдалось

отримати згладження шумів сигналу та в результаті стабільну роботу основного контролера.

Робота з акселерометром представлена наступним чином:

```
annexCode();
while((micros()-timeInterleave)<INTERLEAVING_DELAY) ; //interleaving delay between 2 consecutive reads
  timeInterleave=micros();
  ACC_getADC();
  getEstimatedAttitude(); // computation time must last less than one interleaving delay
  while((micros()-timeInterleave)<INTERLEAVING_DELAY) ; //interleaving delay between 2 consecutive reads
  timeInterleave=micros();
  f.NUNCHUKDATA = 1;
  while(f.NUNCHUKDATA) ACC_getADC(); // For this interleaving reading, we must have a gyro update at this point (less delay)
  for (axis = 0; axis < 3; axis++) {
    // empirical, we take a weighted value of the current and the previous values
    // /4 is to average 4 values, note: overflow is not possible for WMP gyro here
    gyroData[axis] = (gyroADC[axis]*3+gyroADCprevious[axis])/4;
    gyroADCprevious[axis] = gyroADC[axis];
  }
```

Результат виконання програми представлений на рисунку 3.16.

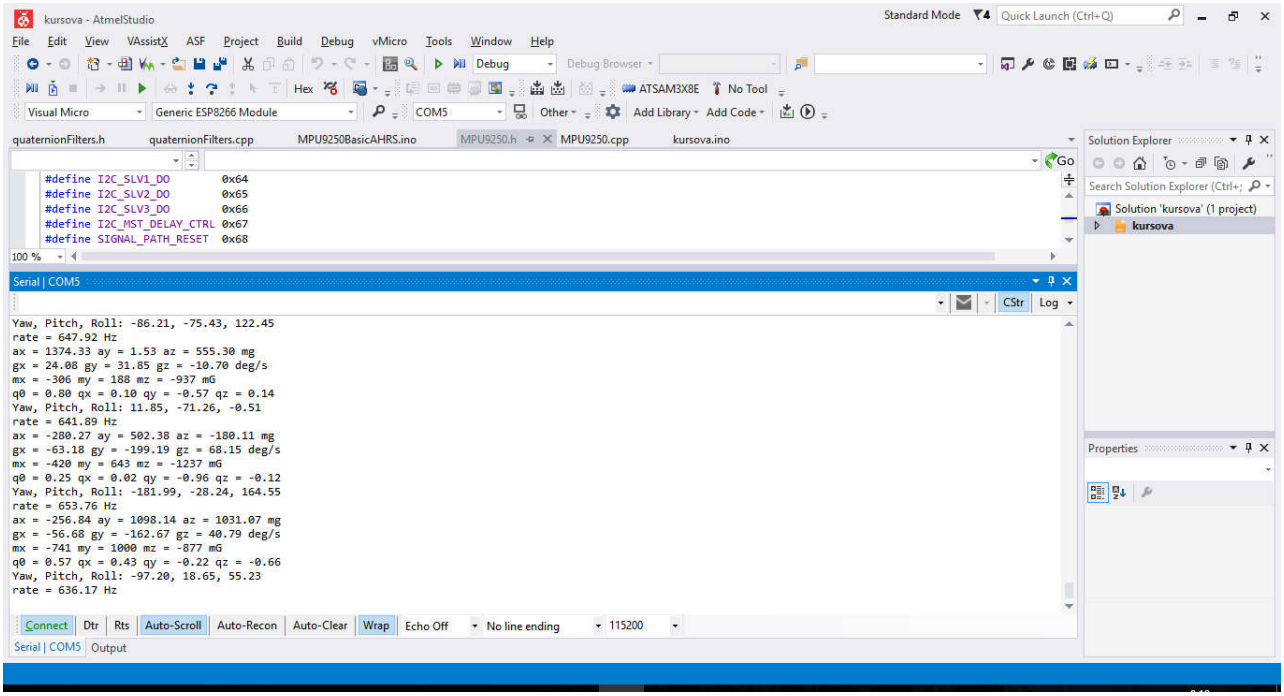

Рисунок 3.16 - AtmelStudio 7 результат виконання програми – показники датчиків в human readable форматі.

Дана функція повертає нам відстань між двома точками в сантиметрах.

```
void GPS_distance_cm_bearing(int32_t* lat1, int32_t* lon1, int32_t* lat2, int32_t* lon2,uint32_t* dist, int32_t* bearing) {
  float dLat = *lat2 - *lat1;       // difference of latitude in 1/10 000 000 degrees
  float dLon = (float)(*lon2 - *lon1) * GPS_scaleLonDown;
  *dist = sqrt(sq(dLat) + sq(dLon)) * 1.113195;
  *bearing = 9000.0f + atan2(-dLat, dLon) * 5729.57795f;
                                        // Convert the output redians to 100xdeg
  if (*bearing < 0) *bearing += 36000;
}
```

Графічний конфігуратор написаний на processing, являється кросплатформеним Java додатком (рисунок 3.17). На даний момент скомпільовано для Linux, Windows та MacOS операційних систем 32х та 64х бітних архітектур.

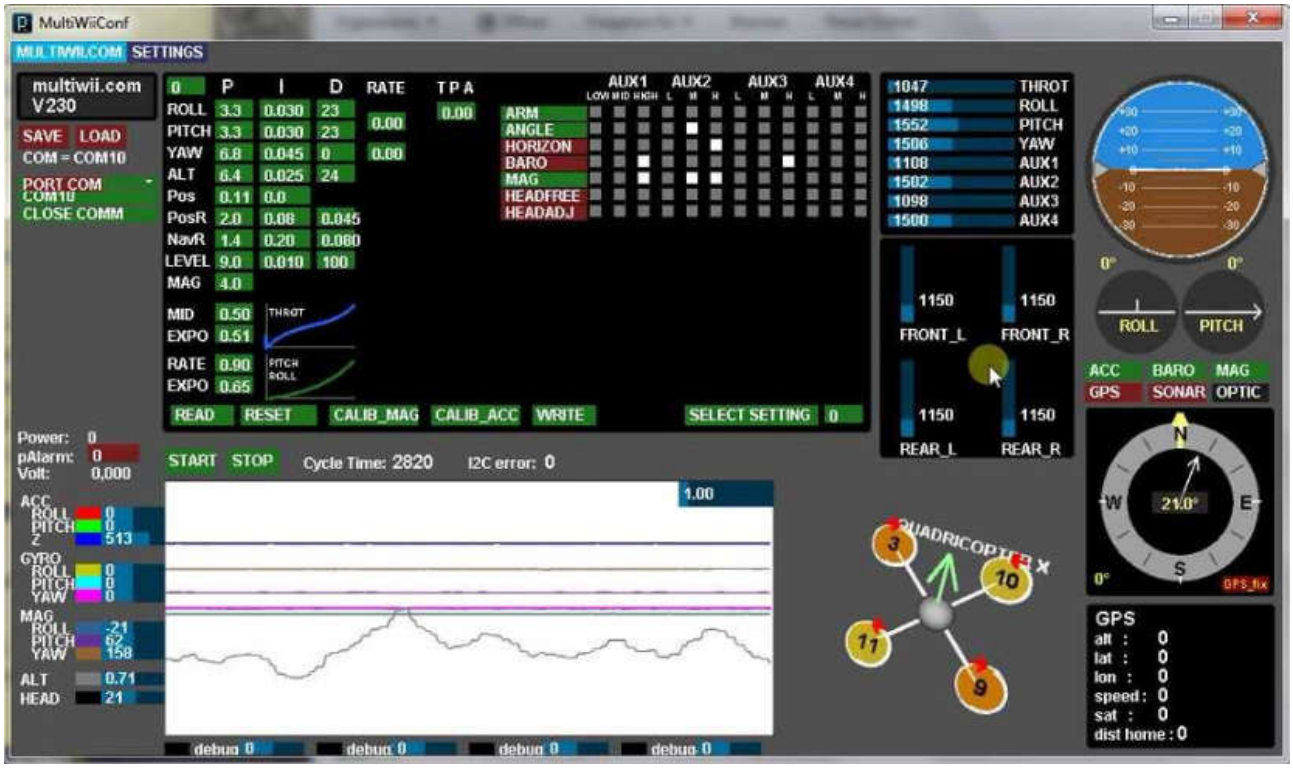

Рисунок 3.17 – MultiWiiConf показники сенсорів.

Даний конфігуратор призначений для зміни та коректування налаштувань польотного контролера, відображення поточних показників сенсорів. У розділі комунікаї доступний список COM портів компютера для підключення контролра (рисунок 3.18).

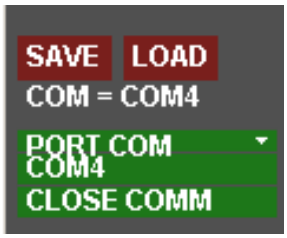

Рисунок 3.18 – Розділ комунікації.

Також у даному розділі можливо провести завантаження попередньо збереженого конфігураційного файлу, та збереження поточних налаштувань системи.

У розділі параметрів та налаштувань можливо змінювати режими роботи контролра та коректувати реакцію на зовнішні команди (рисунок 3.19).

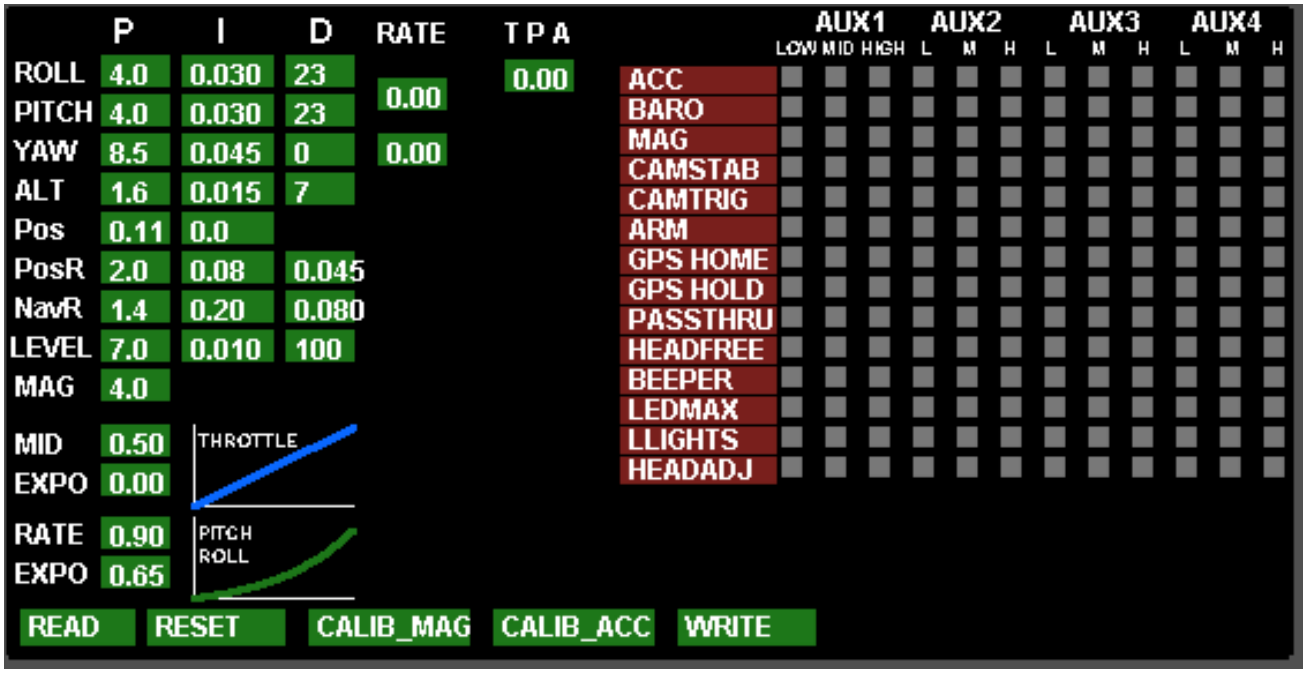

Рисунок 3.19 – Розділ конфігурації режимів роботи.

У розділі дебаг на графіку відображаються в реальному часі показники сенсорів бортової системи квадрокоптера. Одна з кривих відображає показники що надходять з контролера інфрачервоних дальномірів (рисунок 3.20).

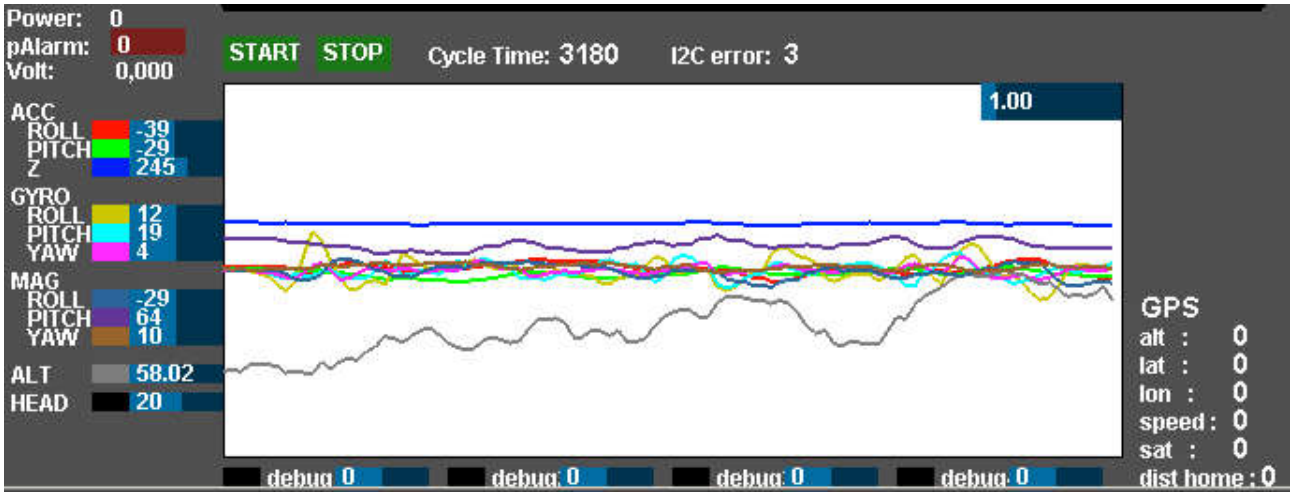

Рисунок 3.20 – Розділ показників сенсорів.

По праву сторону розположено тип моделі квадрокоптера, а також вказівники компасу та нахилу квадрокоптера. Зверху над графіком розміщені налаштування PID регулятора, активації режимів та деякі статусні дані. Для того щоб отримати поточні дані потрібно натиснути кнопку Read, після

внесених змін дані записуються в память контролера за допомогою кнопки Write.

Візуалізація отриманих даних з інфрачервоних сенсорів відстані та контролера польоту MultiWii представлена на рисунку 3.21. Дана візуалізація дозволяє отримати локальну карту перешкод. В результаті можливо коректувати режими роботи контролера польоту та вносити корективи в його траєкторію переміщення.

Одночасна локалізація і картографування (SLAM) є алгоритмічною обчислювальною задачею побудови і оновлення мапи невідомого оточення з одночасним відстежуванням місцезнаходження рухаючись по ньому.

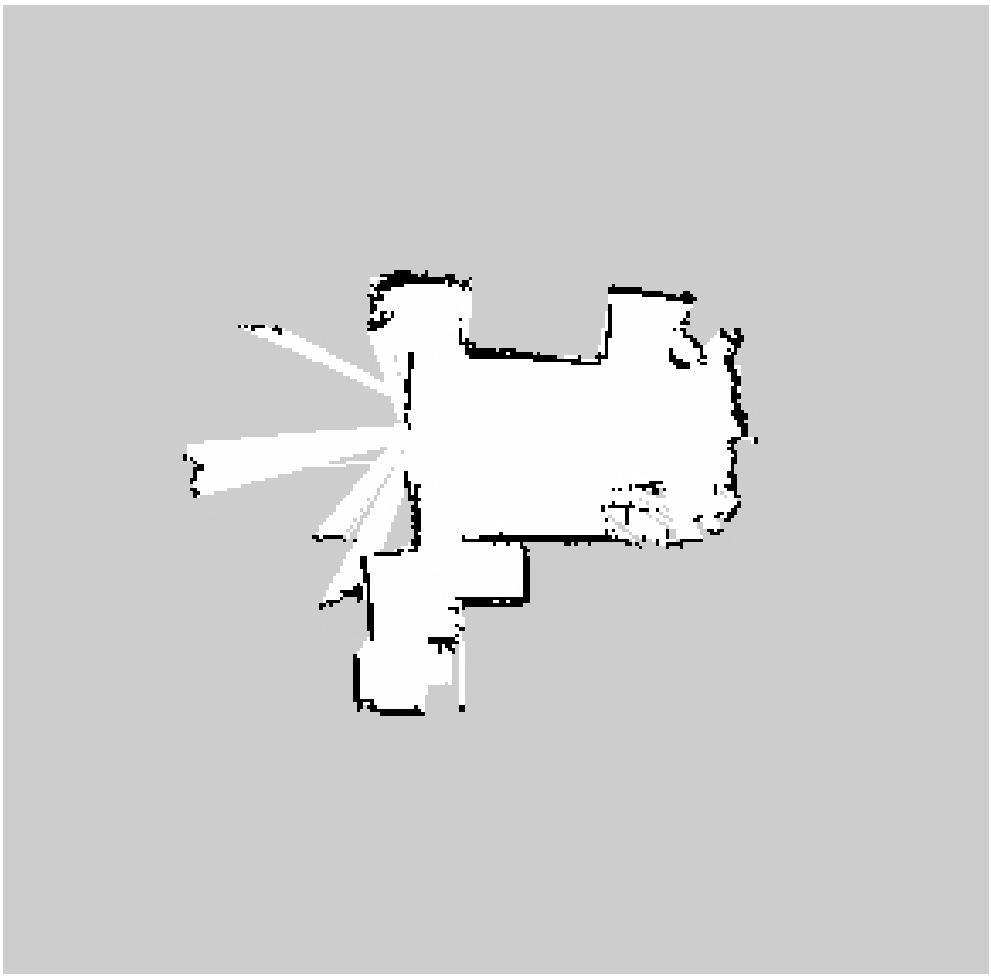

Рисунок 3.21 – Візуалізаці даних інфрачерфоних сенсорів відстані.

Нові алгоритми SLAM досі потребують активного дослідження і пошуку, часто обумовлений різними вимогами і припущення щодо типів карт, датчиків і

моделей. Більшість SLAM систем можна розглядати як комбінації виборів кожного з аспектів.

Під час тестування у лабораторних умовах система навігації роботехнічної платформи показала хороші результати в позиціонування та орієнтації у просторі, побудові траєкторій, керування платформою на швидкості від 0.01 до 50 км/год. За результатами тестування помилок у роботі модуля не виявлено.

ВИСНОВКИ

У даній дипломній роботі було розроблено програмно – апаратний модуль для навігації роботизованої системи на основі інфрачервоних давачів. В результаті виконання дипломної роботи:

- Проаналізовано алгоритми функціонування існуючих глобальних супутникових систем навігації та систем навігації які використовуються для локальної навігації.
- Проведено тестування системи на основі алгоритму SLAM, яка використовувалась у магістерській роботі. Нові алгоритми SLAM досі потребують активного дослідження і пошуку, часто обумовлений різними вимогами і припущення щодо типів карт, датчиків і моделей. Більшість SLAM систем можна розглядати як комбінації виборів кожного з аспектів.
- Створено програмно-апаратний модуль з фільтрацією вхідних сигналів з використанням фільтру Калмана для покращення навігації роботизованої системи на основі інфрачервоних давачів Sharp 2y0a21. За результатами тестування помилок у роботі модуля не виявлено.
- Під час тестування у лабораторних умовах система навігації роботехнічної платформи показала хороші результати в позиціонування та орієнтації у просторі, побудові траєкторій, керування платформою на швидкості від 0.01 до 50 км/год.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шмальцер Р.В. Система навігації робототехнічної платформи за допомогою ультразвукових сенсорів /Р.В. Шмальцер, А.Р,Боднар // Сучасні інформаційні технології 2015: матеріали п'ятої міжнародної конференції студентів і молодих науковців. - Одеса: ОНПУ, 2015. - с. 187-188Дэвид А. Паттерсон Архитектура компьютера и проектирование компьютерных систем / Паттерсон Дэвид А., Хеннесси Джон Л.– Питер, 2012. – 784 с.

2. Жерновий Ю. В. Імітаційне моделювання систем масового обслуговування: Практикум / Ю. В. Жерновий – Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. – 307 с.

3. Раскин Дж. Интерфейс: новые направления в проектировании компьютерных систем / Раскин Дж. — Символ-плюс, 2005. – 272 с.

4. Рошан Педжман Основы построения беспроводных локальных сетей стандарта 802.11 / Рошан Педжман, Лиэри. Джонатан :Пер.с англ.- М.:Издательский дом «Вильямс»,2004.-304 с.

5. Скляр Бернард. Цифровая связь. Теоретические основы и практическое применение,2-е издание / Скляр Бернард: Пер. с англ.- М.:Издательский дом «Вильямс», 2003.-1104с.

6. Таненбаум Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. Ван Стеен – СПб.: Питер, 2003

7. Методичні рекомендації до виконання курсового проекту з освітньо-кваліфікаційного рівня «Магістр» напряму підготовки 6.050102 “Комп’ютерна інженерія” фахового спрямування “Комп’ютерні системи та мережі” » / О.М. Березький, В.М. Теслюк, Р.Б. Трембач, Г.М. Мельник, О.Ю. Борейко, / Під ред. О.М. Березького. - Тернопіль: ТНЕУ, 2015.–35 с.

8. Практическое программирование Arduino/Freeduino. [Електронний ресурс] / Режим доступу: http:/robocraft.ru/blog/arduino/84.html

9. Евстифеев А.В. «Микроконтроллеры AVR семейства Mega» / А.В. Евстифеев /– М.: Издательский дом «Додэка - XXI», 2007.-595с.

10. Бессарабов Б.Ф. Справочник "Диоды, тиристоры, транзисторы и микросхемы широкого применения"/ Б.Ф. Бессарабов, В.Д. Федюк, Д.В.Федюк / - Изд. «Воронеж», 2003-320с.

11. Бобровский С.Н. Навигация мобильных роботов / С.Н. Гончаров, С.Н. Бобровский / Журн. PC Week. - 2004. - №9. - С. 60-63.

12. Мартыненко Ю. Г. Управление движением мобильных колёсных роботов [Текст] / Ю.Г. Мартыненко / МГУ им. М.В. Ломоносова, 2005. - 29-80с.

13. Накано Э. Введение в робототехніку. / Э. Накано / - М.: “Мир”; 1988. - 335с.

14. Юревич Е. И. Основы робототехники. – 2-е изд., перераб. и доп. / Е. И. Юревич /– СПб.: БХВ-Петербург, 2005. – 416с.: ил.

15. Жимарши Ф. Сборка и программирование мобильных роботов в домашних условиях /Ф. Жимарши./ – СПб.: NT Press, 2008. – 410 с.

16. Вильямс Д. Программируемые роботы / Д. Вильямс. / – СПб.: NT Press, 2006. – 311 с.

17. Бишоп О. Настольная книга разработчика роботов / О. Бишоп./ – Воронеж: Корона-Век, 2010. – 207 с.

18. Аналоговые сенсори Arduino. [Електронний ресурс] / Режим доступу: http:/mk90.blogspot.com/2009/07/arduino.html – Назва з титул. екрану.

19. Практическое программирование Arduino/Freeduino - [Електронний ресурс]/ Режим доступу: http:/robocraft.ru/blog/arduino/84.html – Назва з титул. екрану.

20. Евстифеев А.В. «Микроконтроллеры AVR семейства Mega» / А.В. Евстифеев /– Москва – Издательский дом «Додэка - XXI», 2007.-595с.

21. Чернышев А.Ю. Электронная и микропроцессорная техника: учебное пособие / А.Ю. Чернышев, Е.А. Шутов / – Томск: Изд-во Томского политехнического университета, 2010. – 135 с.

22. Евстифеев А.В. «Микроконтроллеры AVR семейства Mega» / А.В. Евстифеев / – Москва – Издательский дом «Додэка - XXI», 2007.-595с.

23. Allan A. «Distributed Network Data» / A. Allan, K.Bradford / O'Reilly Media, Inc., 2013. — 168 pages.
24. Anderson R. Pro Arduino (+source code) / R. Anderson., D. Cervo / Apress, 2013. - 305 p.
25. Application Note: Event-Driven Arduino. Programming with QP: Document Revision H Quantum Leaps, LLC, July 2013. - 34 p.
26. Barrett S.F. Arduino Microcontroller Processing for Everyone Synthesis Lectures on Digital Circuits and Systems. / S.F. Barrett / Morgan & Claypool Publishers, 2010. — 344 p.
27. Bayle J. C. Programming for Arduino / J. C. Bayle / Packt Publishing, 2013. — 512 p.
28. Böhmer M. Beginning Android ADK with Arduino / Böhmer M. / Apress. 2012.- 310 c.
29. Boxall J. Arduino Workshop: A Hands-On Introduction with 65 Projects / J. Boxall / - No Starch Press, 2013. — 394 p.
30. Evans B. Beginning Arduino Programming / B. Evans / - Apress, 2011. - 270 p.
31. Evans M. Arduino in Action / M. Evans, J. Noble, J. Hochenbaum / - Manning Publications Co., 2013. — 368 p.
32. Goransson A. Professional Android Open Accessory Programming with Arduino / A. Goransson, D. Ruiz / - John Wiley & Sons, Inc., 2013. - 408 p.
33. Igoe T. Getting Started with RFID: Identify Objects in the Physical World with Arduino / T. Igoe / O'Reilly Media, 2012. - 42 c.
34. Karvinen T. Make a Mind-Controlled Arduino Robot / T. Karvinen, K. Karvinen / Brain as a Remote O'Reilly Media, 2011. - 96 p.
35. Kelly J.F., de Vinck M. MintDuino: Building an Arduino-compatible Breadboard Microcontroller / J. F. Kelly, M. de Vinck / Maker Media, 2011. — 58 p.

36. Kelly J.F. Arduino Adventures: Escape from Gemini Station / J.F. Kelly, H. Timmis / Apress, 2013. - 332 p.
37. Lazar J. Arduino and LEGO Projects / J. Lazar / - Apress, 2013. — 202 p.
38. Monk S. 30 Arduino Projects for the Evil Genius + Samples / S. Monk / McGraw-Hill/TAB Electronics, 2010. - 206 p.
39. Monk S. Programming Arduino Next Steps: Going Further with Sketches / S. Monk/ McGraw-Hill:, 2013. - 288 p.
40. Randofo P. 20 Unbelievable Arduino Projects / P. Randofo / - Instructables E-book Collection - 430 p.